TAMPERE UNIVERSITY OF TECHNOLOGY

Study of Negative effects of Traffic Localization

Master of Science Thesis

César Mira Sánchez

Department of Communication Engineering
Tampere University of Technology
May 2, 2011

Tutor: Dmitri A. Moltchanov
Examiner: Dr Yevegni Koucheryavy
Examiners and topic approved in the Computing and Electrical Engineering Faculty Council meeting on 8 June 2011

Tampere, May 2, 2011. César Mira Sánchez

Abstract

TAMPERE UNIVERSITY OF TECHNOLOGY
Department of Communication Engineering
MIRA SANCHEZ, CESAR: "Study of Negative effects of Traffic Localization"

Master of Science Thesis, 55 pages.
June 2011
Major subject: Communications Engineering.
Examiners: Dr. Dmitri Moltchanov and Dr. Yevgeni Koucheryavy.
Keywords: Traffic Localization, P2P, Oversim, OMNeT++, Linux.

P2P networks has become one important network for users and ISP. Users wants to share and take profit of this new networks. On the other hand, ISP don not want users' to use so intensively their internet connections because their profits are being reduced.

Traffic Localization has been announced as a solution for P2P disadvantages. It reduces the traffic exchanged between users making cluster. Only a few users from one cluster are going to change data to other networks. There are several studies that indicates the benefits of this measure but there are not too much studies about negative effects.

In our work we tried to simulate a BitTorrent network. Once we have this network ready we constructed it making clusters simulating some Traffic Localization technique. Making several simulations we want to prove how traffic localization affects users' experience.

Acknowledgments

I wish to express my deep gratitude to Professor Dmitri for their availability and support during the project. I especially appreciated the information and advice you offered me. Also, I want to thank the Communication Engineering Department and the Tampere University of Technology for giving me the chance for do this project.

Also I want to express my gratitude to my parents, for all the support that they gave me and are still giving me.

Also, I want to mention my friends and family for helping me during all this time.

Again, thank you so much for everything. I greatly appreciate your generosity.

Abbreviations

- ADSL: Asymmetric Digital Subscriber Line.
- ARPANET: Advanced Research Projects Agency Network.
- AS: Autonomous System.
- C/C++. Programming language.
- DHT: Distributed Hash Table.
- DRM: Digital Rights Management.
- DSL: Digital Subscriber Line.
- eD2k: eDonkey2000.
- EDGE: Enhanced Data rates for GSM Evolution.
- FAQ: Frequently Asked Questions.
- GNU: Compiler Collection.
- GSM: Global System for Mobile Communications.
- GUI: Graphical User Interface.
- HDD: Hard Disk Drive.
- HTTP: Hypertext Transfer Protocol.
- IANA: (Internet Assigned Numbers Authority).
- IETF: Internet Engineering Task Force.
- Inter/Intra AS (Inter/intra Autonomous System).
- IP: Internet Protocol.
- IPv4: Internet Protocol version 4 (RFC 791).
- IPv6: Internet Protocol version 6 (RFC 2460).
- ISO: form standard ISO 9660.
- ISP: Internet Service Provider.
- ITU: International Telecommunication Union.
- Java. Programming language.

- LAN: Local Area Network.
- MP3: MPEG-1 Audio Layer III o MPEG-2 Audio Layer III.
- NAT: Network Address Translation.
- P2P: Peer To Peer.
- RFC: Request For Comments. memorandum published by the Internet Engineering Task Force (IETF).
- Ping: utility used to test the reachability of a host.
- PSTN: Public Switched Telephone Network.
- RIAA: Recording Industry Association of America.
- SGAE: Sociedad General de Autores y Editores.
- SMTP: Simple Mail Transfer Protocol.
- TCP: Transport Control Protocol.
- UDP: User Datagram Protocol.
- UMTS: Universal Mobile Telecommunications System.
- VoIP: Voice Over Internet Protocol.
- WAN: Wide Area Network.
- WWW: World Wide Web.

List of Figures

Contents

1 Introduction

Throughout Human History there have been many inventions that have changed in some way the evolution of humanity. Some of those changes have been the fire, the wheel or electricity. We are now attending the most recent great discovery that has been introduced into our lives. That is the Internet.

With the fire we learned how to keep warm, how to cook, how to defend us from wild animals. With the wheel we learned how to carry heavy things easily, it made us possible trading and traveling. With electricity we discovered a new magnetic world, electromagnetism, etc. Internet, the last great discovery, has changed in many ways our consumption habits and our behavior, even in the way how we relate to others.

Internet has become a universal service. Some years ago, we could not imagine the importance that this kind of new technology would take into our lives. Nowadays, we can do almost everything on the Internet. Daily activities as shopping, making telephone calls (VoIP), videoconferences or watching videos are now instantly resolved online: sharing our work with other institutions, knowing people, buying tickets, watching online maps, voting, sharing experiences and pictures, etc. Internet possibilities' are still unimaginable.

Internet was created back to 1969, when the first connection of computers - ARPANET- was established between three universities in California and another one in Utah, USA. In 1972 the first public demonstration of ARPANET was performed. It was a new communications network funded by DARPA that worked distributed over the PSTN. In 1983, ARPANET changed protocol NCP to TCP / IP. That same year, the IAB was established to standardize the TCP / IP and to provide research resources to the Internet. On the other hand, focused allocation function in the IANA identifiers later delegated some of its functions in the Internet registry, which in turn provides the DNS services. In 1989, with the integration of OSI protocols in the Internet architecture, started the current trend of allowing not only the interconnection of disparate structures, but also to facilitate the use of different communication protocols.

Internet is a dynamic network that has increased and is still increasing its size exponentially. An example of the evolution of the Internet is the number of computers connected. When the Internet was created there were only 4 computers. Nowadays it is impossible to determinate exactly the number of electronic devices (including computers, laptops, routers, TVs, smart-phones, etc.) that are part of the network.

One possible approximation to determine the magnitude of Internet is with the number of existing IP addresses. An IP address is a numeric tag that identifies, in a logical and hierarchical way, an interface (an element of communication / connection) of a device (usually a computer) in a network that uses IP (Internet Protocol), which corresponds to the network layer of TCP / IP.

IPv4 addresses are represented by a 32 bits binary number. IP addresses can be expressed as decimal numbers by dividing the 32-bit address into four octets. The decimal value of each octet is from 0 to 255. There are some addresses reserved for special purposes such as private networks (around 18 million addresses). One example is the typical private address 192.168.0.1. Another reserved addresses are for multicast (around 270 million addresses). Also, there are some reserved address like 127.x.x.x. This address is reserved to denote the machine itself. It is called loopback.

According to the IANA (Internet Assigned Numbers Authority connection) the number of IPv4 were finished on February 2011. There are 4.294.967.296 directions according to the version 4 of the Internet Protocol. This number does not respond to the number of users, but it shows anyway the size of this network.

In computer networks, a subnet is a logical address range. When a computer network becomes large, it is divided into subnets for the following reasons:

- Reduce the size of broadcast domains.
- Make the network more manageable, administratively. Among others, you can control traffic between different subnets, using ACLs.

A node that uses TCP/IP protocol has an IP address with a subnet mask associated. The subnet mask indicates which bits (or what portion) of its address identify the network. The mask consists of a sequence of ones followed by a sequence of zeros written in the same way that an IP address, for example, a 32-bit mask 255.255.240.0 would be written, i.e. a 20-bit IP address followed by one 12 bits to 0, but separated in blocks of 8 bits written in decimal. The mask determines all the parameters of a subnet, network addresses, broadcast addresses (broadcast) and addresses assigned to network nodes (hosts).

Nowadays, we have to migrate and start using IPv6 to be able to assure one IP direction to every device wanting to get connected to the Internet. The Internet Protocol version 6 (IPv6) is a version of Internet Protocol (IP), defined in RFC 2460 and designed to replace Internet Protocol version 4 (IPv4) RFC 791, which is currently implemented in most devices that have access to the Internet.

Internet is made of multiple networks and has a decentralized structure. It was created with the idea that if any part suffered a total collapse, the messages would find the way to reach its destination anyway. Some important elements on the internet are:

- Routers: A router is a device that forwards data packets across computer networks.

- Server: Computer program running as a service, to serve the needs or requests of other programs (clients) which may or may not be running on the same computer.
- DNS: it translates domain names meaningful to humans into the numerical identifiers.
- Last mile: connection between users equipment and ISP equipment.
- Connections: connects all ISP equipment around the world (for example, fiber optic, wireless routers, coaxial wires,etc.)
- Terminal and Client: Users equipment to connect to the internet. We need one application (for example, web browser) to be connected.

Internet is based on a set of protocols. There are over 100 different. The two most important protocols that compose it are TCP (Transmission Control Protocol) and (IP) Internet Protocol, which were the first two to be defined, and are the most used. Among other popular protocols there is HTTP (HyperText Transfer Protocol), which is used to access web pages, plus others such as ARP (Address Resolution Protocol) to resolve addresses, FTP (File Transfer Protocol) for file transfer, SMTP (Simple Mail Transfer Protocol), POP (Post Office Protocol) for email, TELNET to access remote computers, among others.

TCP / IP is the foundation of the Internet and is used to link computers running different operating systems, including PCs, minicomputers and mainframe computers over local area networks (LAN) and wide area (WAN).

One of the networks that has become very important on the Internet, according to traffic volume and user utilization is P2P networks. We have multitude of P2P networks, for example: Gnutella, BitTorrent [3], eDonkey, Napster, Ares, etc. The basic idea of this kind of networks is that every user is at the same time client and server. This way, the user is going to download and upload a file simultaneously.

Nowadays, P2P networks have many utilities. Commercial and non-commercial organizations use this kind of network for its own profit. One example of non-commercial company using P2P networks is Ubuntu. When they have just released the latest version of their own operating system they recommend to use P2P network instead of the normal server (many users can overload the server). Another example could be every person wanting to share some files with some other people. One example of a commercial company using P2P technology is Spotify. Every user downloads the songs he wants to listen from other users instead of everyone from the central servers. P2P is one possible solution for small companies or organizations to create large networks with small resources.

Here are some characteristics of P2P according to the [15] data of Internet Study 2008/2009:

- P2P generates most traffic in all regions.
- The proportion of P2P traffic has decreased but it remains still as the number one of all protocols, HTTP is the second one.
- The proportion of eDonkey is much lower than last year.
- File hosting has considerably grown in popularity.
- Streaming is taking over P2P users for video content.
- Usenet, a file sharing alternative for P2P, appears in the statistics for the first time.

There are many reasons of the success of this kind of networks. One of these reasons is the easy process to share content that we have in our own computers with as many users (or clients) as are on the Internet. Another of the reasons is the scalability. In P2P systems, when the number of clients increases, the total capacity of the system also increases. On the other hand, in a typical client/server architecture, clients do not share their resources, so if the total number of clients increases, the capacity of the system decreases. Another reason is the storage space. In conventional client/server architecture we have limited space for sharing documents, images, etc. In P2P network, the limit is client's hard disk device. Another reason of the success of some P2P networks is the amount of free multimedia data available (music, movies, books,etc.). Usually connect and download something of this kind of networks is free.

But, not everything is positive in P2P networks. As we just read, in a P2P network every client is going to act as a client and also as a server. So if we compare the bandwidth consumed for a user that uses P2P networks with another one that only uses client/server services we can appreciate that the first ones are using the network more than the average. ISP construct their infrastructure according to some estimations of usage. If most of the users use their connections more than the estimated, it can produce technical problems of capacity.

In addition to the problem of increase of bandwidth, most of the new extra traffic generated by P2P system passes through third networks. This increase means that operating costs of user company increases too. ISP had agreements with other companies or/and another ISP. More traffic though other networks is harmful for this agreements. This is one of the main reasons why ISP oppose to P2P networks.

There are numerous studies and mechanisms to minimize or reduce the negative effects of P2P networks. As we will see further in chapter 3, one of the methods used is “Traffic Localization”. There are studies that show us how to decrease the interAS traffic in P2P network. Using different techniques, as commented in [7].For example, using a external element called “Oracle” to address users. Also, we can use some algorithms to match peers that are in the same network, country.

In our work we are going to study how these new techniques of “Traffic Localization” may affect the final user experience. How time delay increases or decreases if we use some technique of “Traffic Localization”. We chose BitTorrent protocol for our experiments because now a days, is the most important network and protocol. For making this possible we are going to choose, install and configure a BitTorrent simulator. Then we are going to modify it for create clusters and study network behavior.

2 P2P Technologies

P2P (Peer-to-Peer) computing or networking is a distributed architecture application that partitions tasks or workloads between peers. All the peers are equally privileged and have the same function in the network. P2P protocols operate at the application layer of the TCP/IP model.

P2P networks and protocols were very unusual in the beginnings of 2000. The most common architecture was client/server, for example, FTP, HTTP, SMTP, etc. P2P network was born as a revolution of conventional networks architecture.

There are many utilities for P2P networks. The most known are used to share files (audio, video, text, software, ISO images etc.). But there are many others: VoIP (Voice Over IP, Internet telephony systems), distribution of audiovisual content and TV programs, scientific calculations that process large databases also to facilitate the exchange of information in medical research and exchange of files in academia. The potential of P2P networks is almost unlimited.

Most of P2P network protocols have become very popular. They have several "legal" problems and are forced to close or change their functionality in order to continuing existing. For example, Napster was demanded by artists like Metallica, Madonna, Dr. Dre in the beginnings of 2000 when they founded that some of their new singles were available for free in Napster, some times even before they published it. We can say that P2P networks facilitation of transfer of copyrighted material raise the ire of the Recording Industry Association of America (RIAA) and artists that can see how easy is distributed his job in some of this networks.

For telecommunications companies, the rise of P2P networks represented an increase in costs and use of its own infrastructure and also to third networks. As we are going to see, the fact that a user acts as a server and client increases the traffic between peers and often traffic between different networks. In this case, ISP has to renegotiate with other ISP or companies new traffic exchange agreements.

2.1 P2P technology

2.1.1 P2P Topologies and Classification

Topology is defined as the chain of communication used by the nodes that form a network to communicate. This chain of communications can be physical or logical. The physical structure does not have to agree with the logic and vice versa. Some examples of topologies are line, ring, mesh, fully connected, bus, tree, star, etc.

Here is a graphic example of different topologies:

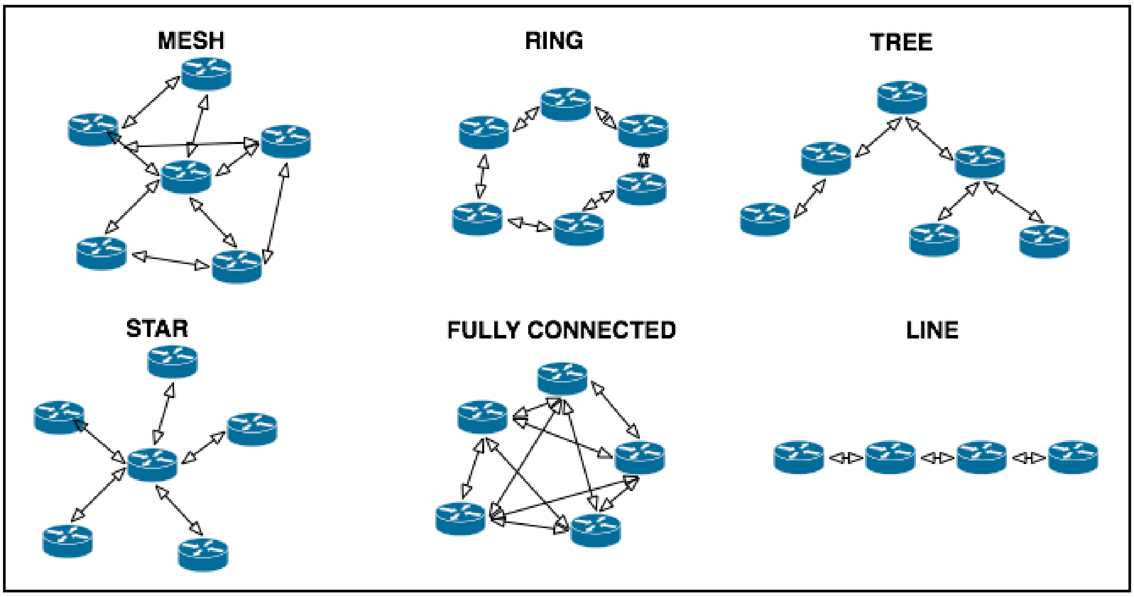

Figure 1: Network Topologies

Not all the P2P networks have the same topology and it is very difficult to determinate or classify one P2P network only for its topology. For example, a fully connected topology is almost impossible in networks that are formed by thousands clients or nodes. In P2P networks we have to take in mind that nodes are users connected with their computers, so they can connect and disconnect several times during a day. So it is another reason that makes very difficult to determinate the topology of P2P network. Network topology changes as peers connect/disconnect.

Another way to classify the P2P networks [25] is their degree of centralization. We can classify P2P networks can be centralized, half-centralized (also known as mixed or hybrid) and totally decentralized.

Centralized networks: The network is governed by a single server, which serves as a link between nodes and a server to access the content. The traffic circulates at the request of the nodes. Without the existence of the central server there are no communications. A couple of examples of this type of network are Napster and Audiogalaxy.

Half-centralized networks: In this type of network, we can observe the interaction between a central server that works as a hub and manages the broadband resources, routing and establishing communication between nodes without knowing the identity of each node and without storing any information. As the opposite of what happens with centralized networks, in half-centralized networks if the central server is disconnected data can be exchange between peers/nodes. Some examples of hybrid P2P network are BitTorrent, eDonkey and Direct Connect.

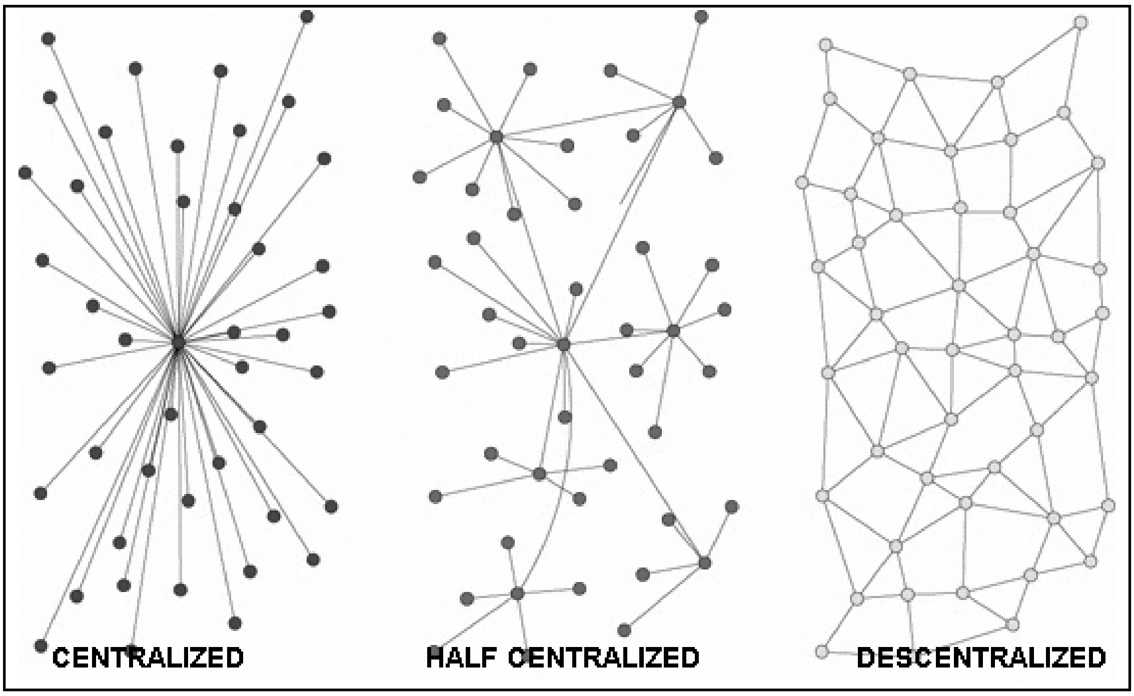

Figure 2: Network Classification

Decentralized networks: This type of networks are the most popular. All communications are directly from peer to peer with the aid of a node (another peer/user) who can link these communications. Some examples of a “pure” P2P network are: Kademlia, Ares Galaxy, Gnutella, Freenet and Gnutella2.

The degree of centralization in a network is very important for users’ experience and for the development of the network. Here are some pros and cons of centralized, half-centralized and decentralize networks.

- Centralized structures are limited for central server capacity on the other hand decentralized networks have almost unlimited capacity.
- Centralized networks are more fragile to network problems.
- Decentralized networks are more complicated to manage.
- In centralized and half-centralized networks when the central server is offline we can not start downloading anything. In Half-centralized networks if the central server goes down during the process of downloading, users can continue working.
- In decentralized networks one petition of search has to cross through all the network. Some of this petitions can be lost.

2.1.2 P2P File Sharing

Now a days, P2P file sharing is the most important P2P application. Also, is the one that more resources consumes. It is very difficult to estimate the percentage of all P2P traffic generated due to file sharing. Many studies show how important is P2P traffic in all data consumed [26]. Few years ago, P2P percentage of traffic was up to 50%, followed by video streaming, web services and others (SMTP, FTP, etc.).

P2P file sharing was popularized due to a software and network called Napster. Napster made relatively easy for music enthusiasts to download copies of songs that were otherwise difficult to obtain; for example older songs, unreleased recordings, and songs from concert bootleg recordings. Some users felt justified in downloading digital copies of recordings they had already purchased in other formats, like LP and cassette tape, before the compact disc emerged as the dominant format for music recordings. Napster was released as a centralized unstructured peer-to-peer system. It requires a central server for indexing and peer discovery. Later, in 2000 other networks like Gnutella, eDonkey2000, FreeNet appeared. And some years later Kazaa, BitTorrent, Ares Galaxy, etc.

To understand the boom P2P networks file sharing had, first we have to understand users' situation few years ago. When P2P networks started to become popular, a user with normal resources could not afford the service of hosting that could permit storage of files up to 100 MB. With this kind of "new" networks, a regular user, just had to save the file that he/she wanted to share with other people on his/her hard drive. The client (Napster, eMule, eDonkey200, BitTorrent) will upload the file. Also, other peers that wanted to download the file would contribute on the process of uploading. So, with a regular connection and basic knowledge of computer science, anyone could transmit large files to as many people wants. Moreover, if more people download one file, more easy will be distributed. Another important reason was the copyrighted and old media files that exists in some P2P networks. With P2P networks, people start to share their own multimedia files: discographies, songs, movies, etc. In some cases, premiers of band groups or movies can also be found in this networks. Other users could download those files for free.

P2P file sharing programs have evolved in many aspects. Not every P2P file sharing program use the same protocol. Also, the structure of P2P programs is different. Some of them have been written from the beginning and others are a evolution that existing ones (for example, BitTorrent is based on eDonkey2000 protocol). The structure has also evolved. Earlier versions of P2P networks evolved from centralized structures to unstructured ones. Now a days, P2P networks implement a direct, encrypted and anonymous connection.

From the development of Napster through Kazaa (FasTrack protocol), until BitTorrent many protocols have been created. Some of them are open source and others are closed and proprietary code. For example, Skype protocol is closed and proprietary code. On the other hand, Ares Galaxy is open source. Nowadays, we can say that the two most popular P2P file sharing protocols are eDonkey2000 (with eMule as the most important software client) and BitTorrent (with BitTorrent protocol and many software clients).

The procedure to use this networks depends on the software and network we want to connect. First of all we have to install the software we are going to use depending on the network we want to connect. For example, if we want to connect to eDonkey2000 network we can use eMule software; if we want to connect to BitTorrent network we can use uTorrent, Transmission, Vuze, etc.

For downloading something from this networks we have two choices, use some search engine or be redirected for some external link. Some networks (eDonkey2000, Ares) incorporate search engine to look for one specific file. This characteristic is present in most of the P2P file sharing programs except, for example, BitTorrent. Users have to put what they are looking for and they will have the results in the same software. This is the most common engine for find material in P2P network. Also, there are other procedures to identify, find and start downloading a file; through an external link. eDonkey2000 incorporate url direction with format *ed2k://—file—name of file—14997504—*. This links are similar than hyperlinks in HTTP protocol. *eD2k* links allow a file to be identified from a link in a web browser and to be downloaded thereafter by a client like eMule, Shareaza or any other compatible software. Users only have to *click* this link and the software will interpret it and start downloading it. Nowadays Magnet links have replaced *eD2k* links in practice. They serve a similar role, but are not limited to the eD2k hash and can contain other hashes such as SHA-1 or MD5 for example, which makes it possible to use these links to point to files in several networks (as well as in BitTorrent) at once. BitTorrent incorporates an external file called *name_of_file.torrent*. This file contains the information for establishing the communications between users and tracker (see point 2.2).

The typical procedure to download a file is very simple. Large files are divided into smaller portions called *chunks*. *Chunks* can be obtained from multiple peers. Users can download chunks from different users and then reassembled in their computer. Small files are directly uploaded to requested peers. This is done while the peer is simultaneously uploading and downloading *chunks*. Here, we can see how peers are simultaneously uploading and downloading the same file.

Once user have all the chunks, software client assemble it. To identify problems in *chunks* many software clients uses Hash functions (algorithms MD5 and SHA-1 are two of the most popular). A Hash function is a function for summarizing or probabilistically identifying a large set of information, resulting in generally smaller than original. A fundamental property of hashing is that if two results of the same function are different, then the two inputs that generated those results are also different but of the same size. For example, if we use Hash function of 128 bits to two files of 10Mb and another file of 100Gb, the results are different and both of 128 bits. Kazaa, Ares Galaxy, Overnet, BitTorrent are examples of clients that use this techniques. Hash functions are also used to identify a record in a database that allow faster access to records (even faster than having index).

During past years, many studies have proved that P2P file sharing is the network that more traffic generate (download and upload traffic). Recent studies indicate that percentage of P2P traffic generated by these networks is decreasing. Other services such as video streaming (YouTube, Vimeo, MegaVideo, etc.) or direct downloads (for example MegaUpload, Fileserve, Filesonic or RapidShare) are gaining importance. In some countries this percentage can be even bigger. But there is a common data in all of them. All the studies assure that P2P file sharing programs generate the most upload traffic of all networks. Even now a days.

As we know, P2P users download information from other users. So, if users does not share P2P is not efficient. Users that once they finish downloading disconnect from the network or move/remove the file are called Free-Riders. Some clients (eMule, BitTorrent) have incorporate a credit system. The credit system is used to reward users that contributes to the network, for example, transferring information to other customers. eMule is one example of credit system. The strict queue system in eMule is based on how long a user has been waiting for. The credit system provides a “major modifier” for this time out, taking into consideration the data uploaded and downloaded between two clients . The more information exchange to users to another, faster progress has on the client queue. It is calculated by taking into account the amount of data transferred between the two clients. In BitTorrent networks, trackers can save information about the data transferred between users. Then, depending on this ratios tracker can ban one *Free-Rider* from the tracker.

Some characteristic of most common system credits:

- Credits are not stored in users’ equipment. It avoids possible manipulations.
- Credits are from user to user, not globals.
- All customers who transferred are rewarded by the credit system. This does not change if the client supports the credit system or not.

In the beginnings of 2000 there were not too much alternatives for P2P networks. Server/client architectures that provide this features were almost inexistent. We can say that FTP was the most common alternative for file sharing. Now a days, there are plenty of alternatives. For example, large files web hosting like (MegaUpload, RapidShare, FireMedia etc.), FTP, some backup utilities like DropBox and also for small files (less than 20Mb) email. Also, with the modern bandwidth and HDD we can make transfer of medium files from one computer to other in live, for example with Skype software.

For prevent some “ilegal” exchange of copyrighted material in P2P networks, companies (Sony, Microsoft, BBC, Apple, etc.) insert DRM in media files. DRM is a generic term that refers to access control technologies used by publishers and copyright owners to restrict the use of digital media or devices. DRM are being included in all types of digital devices, without informing those who buy for their consequences.

2.1.3 P2P Streaming

P2P is one of the services that produces and consumes more resources, especially band with [17]. As we have already discussed in the introduction of this chapter, latest data on telecommunications networks’ use indicates that there has been an increase in video on demand services such as YouTube, Vimeo, Spotify, TV online, etc. Audiovisual content is growing up on demand and also in quality. Now a days, more TV-channels are emitting in High Definition (HD). There are many reasons for this increase. Now a days, users do not want to wait weeks or even years to watch their favorite TV-show. They prefer to watch it online. Also, Streaming content is usually more easy to use than P2P. Some web pages stores links to video streaming services; users just have to *click* the link and wait for the video. In traditional P2P systems, users have to download the hole file and then watch it.

P2P streaming is a transmission technology of audiovisual content (video, television, etc.) via the Internet using P2P systems architecture, where individual nodes are connected to other nodes to receive video and audio streams rather than through a central server, as in IP-based television (IPTV). P2P media streaming is pretty new technology and there are no de facto standards.

Some existing Peer-to-Peer Media Streaming Technologies are:

- BitTorrent DNA http://www.bittorrent.com/dna/
- Abacast http://www.abacast.com/
- Itiva http://www.itiva.com/
- Mediazone http://www.mediazone.com/index.html
- NiFTyTV Online Television http://www.niftytv.com/
- PPLive http://www.pplive.com/
- SopCast http://www.sopcast.com/
- TvAnts http://www.tvants.com/
- TVU networks http://www.tvunetworks.com/

The main characteristics of video and audio services on demand are:

- High and constant consumption of bandwidth.
- We need low delay.
- High scalability.
- High adaptation on changes on the network (users disconnections).

As we can see in [23], the procedure of P2P Streaming programs is almost the same for every program. First of all, we need a buffer to fill (in some programs, like SopCast and PPLive we have 2 buffers). The main buffer is used to save chunks that we receive from other users. Reproduction will start when the buffer is full. One important parameter in this kind of software and protocols is the size of the buffer. If the size is too big, we are going to spend many time filling it and there will be more delay between the original signal and the received. On the other hand, if the buffer is small we may not have time to fill the buffer and the image will be wrong.

Some considerations about P2P Streaming:

One of the major problems with this technology is the delay. If we want to have a good user experience in this networks we need low ping connection from our computer to destiny computer. Delay in this network - so users' experience- is proportional to users' ping.

Not every internet access is suitable to support P2P live streaming. Connections with high ping characteristic are not recommended. As we can see in [23], EDGE and UMTS lines are not recommended for transmissions up to 350 Kbps. Also lines based on DSL have high PING.

As we have seen, one of the characteristics of P2P networks is that each user shares their bandwidth with other users in the network. Therefore, we can say that P2P networks can be a good support for multimedia content distribution.

Users' localizations is not important. Channels/videos can bee seen world wide. Users only need Broadband connection.

Costs of signal distribution around the world are non-existent compared with traditional systems (TV, Satellite, Cable).

2.1.4 P2P VoIP

VoIP is a specific sets of protocols to carry voice over the IP. It allows to avoid classic PSTN fares. It is also used to make long-distance calls (very expensive with traditional telephone technology), make videoconferences, make multi conferences, chat, etc.

As happens with P2P file sharing networks, VoIP networks have evolved [23] from also centralized structure, where all the voice packets that form the call pass through a central server, to a half-centralized structure where the central server is used only for login and identify the user in the network.

The benefits of P2P VoIP are numerous. Here is a list of some of them:

- We only need a data infrastructure in order to be able to make a call.
- Lower costs since only Internet access is needed.
- Calls can be encrypted.
- Location independent.
- Integration with other services.
- Calls can be done from IP telephone or from a computer.
- Is independent of hardware used.
- Provides a link to the traditional telephone network.

There are three big elements in a VoIP network: client, servers and gateways. Client establishes and terminates voice calls, this information is coded, packaged and transmitted through the microphone (input) user in the same way the information is decoded and played through speakers or headphones. Servers are handling database operations, performed in real time and in one out of it. These operations are accounting, collection, routing, management and service control, registration of users, etc. Gateways provide a bridge of communication between all users, their main function is to provide traditional telephony interfaces with appropriate, which functioned as a platform for users (clients) virtual.

As we are transporting voice (analogical signal) we have to codify it. For do this we use codec. Codecs ensure the compression and encryption of audio or video for later decoding and decompression before user can generate a usable sound or/and image. Depending on the codec used in the transmission, transmission uses more or less bandwidth. The amount of bandwidth used is directly proportional to the quality of the data transmitted.

Here is a list of possible codecs [1]:

1. G.711: bit-rate de 56 o 64 Kbps.
2. G.722: bit-rate de 48, 56 o 64 Kbps.
3. G.723: bit-rate de 5,3 o 6,4 Kbps.
4. G.728: bit-rate de 16 Kbps.
5. G.729: bit-rate de 8 o 13 Kbps.

Most of this systems use SIP (Session Initiation Protocol), an open protocol. It is formally specified in RFC 3261. SIP allows users to interact with other systems so calls can be made between customers of different systems. Other VoIP protocols are: H.323 (ITU-T) [14], SIP (IETF) [12] , Skinny Client Control Protocol (Cisco), Megaco (control protocol).

The most known example of company that uses VoIP technology is Skype. Skype permit users to call and make videoconferences from one computer to other for free. Also, permits to make calls to fixed lines and mobile phones at reduced fares.

Skype has a proprietary encoders, negotiable voice codecs, proprietary signaling and Gateways to/from PSTN/SIP. Also, it can work across NATs and firewalls. This last ability is very useful for users with not too much knowledge of network configuration.

But, Skype is not the only VoIP software. Here is a small list of alternatives to Skype.

- Google Talk.
- Ekiga.
- Gizmo.
- QuteCom.
- iChat.

P2P VoIP has revolutionize the traditional telephony. VoIP permit users to increase the richness of the media allowing more features in the same call with a significant reduction of the costs for ISP and users.

2.2 BitTorrent

BitTorrent protocol was originally based on free software and was created by Bram Cohen. The first client, known as BitTorrent, was also created by Bram Cohen, in October 2002. It was originally set to Python but today you can find clients written in C or Java, for example. Now a day exists many clients for connect to BitTorrent networks.*Azureus*, *BitComet*, *KTorrent*, *uTorrent o Transmission* are examples of BitTorrent clients.

The method used by BitTorrent to distribute files is similar to eDonkey2000 in many aspects. eDonkey2000 is the most important P2P network. Generally the nodes in edonkey2000 network share and download larger amounts of files, reducing the available bandwidth for each transfer. In BitTorrent, transfers are usually very fast because they focus in a small group of files. Besides the eDonkey2000 protocol does not reward users who share a higher bandwidth. However, we must clarify that the most widespread client for the eDonkey, eMule, it incorporates a credit system to reward good users' behavior. Credit system and faster transfers make BitTorrent the most used P2P file sharing protocol. User experience in BitTorrent is very high.

Unlike other P2P networks, BitTorrent does not include a pure file search mechanism. As we are going to explain later, BitTorrent users need to find on their own torrent files required by the protocol. Typically, these files can be downloaded from websites that publish large files (such as GNU / Linux) or from web search indexes (such as The Pirate Bay, Torrentreactor, Torrentboxs, Showrss ,etc.).

Here are some of the main features of the BitTorrent system [23]:

- Integrity checks done at the piece level.

  Corrupt pieces does not corrupt the entire file.
- Simultaneous downloading from multiple sources.
- Utilization of file *.torrent* that contains metadata information of the file.
- Uploading the content while still downloading it.
- High speed data transfer compared with other P2P networks.
- BitTorrent has no search engine.

BitTorrent file distribution system consists of following entities:

1. BitTorrent Tracker: is a special server that contains the information needed for peers to connect to other peers interested in the same file. Initially it is the only way to find what users containing the file you want to download. Trackers can be public or private. Tracker is connected when starting, refreshing, stopping or completing a torrent.

2. Torrent file: static metadata file. Contains information such as hash information, piece length, file name, content length, the URL of the tracker, etc.

3. Seeder: contains the initial and entire file.

4. Web Server: server that contains *.torrent* files.

5. Leechers: Named to users that are downloading the file but they do not have it completed.

6. Peers: Named to all users on the network.

BitTorrent procedures:

User downloads a *.torrent* file from a web server. It contains the information of the file that he wants to download and data about the tracker (IP); this tracker stores details about other peers, chunks, peers status, hash result, etc. Among other information, *torrent* files also contain the address of the tracker that we need to connect to peers joining the swarm (the *.torrent* file is usually very small, its size is around 8-16 KB).

This *.torrent* has to be opened with a client that can interpret that information. Many of them may be used for free. The most popular clients are uTorrent, BitComet, Vuze (formerly Azureus), Transmission. All of them are based on the original BitTorrent protocol, but just some include makes improvements to the protocol. For example: Azureus was the first BitTorrent client to implement such a system through the distributed hash table (DHT) method.

Peers use HTTP protocol for tracker communication. The tracker reports the list of all peers and seeds that possess parts of the file the user wants to download. The tracker is updated with new peer information just entered by the user.

Once the peer knows where to find the necessary parts, this peer communicates with others using TCP or UDP sockets and the file starts downloading to the user's computer. Each part is downloaded automatically and shared with other peers.

BitTorrent can also use DHT for storing peer contact information for *trackerless* torrents. The DHT protocol is based on Kademlia and is implemented over UDP. The DHT protocol consists on: When a node wants to find peers for a torrent, it uses the distance metric to compare the info hash of the torrent with the IDs of the nodes in its own routing table. Azureus was the first BitTorrent client to implement such a system through the distributed hash table (DHT) method. An alternative and incompatible DHT system, known as Mainline DHT, was later developed and adopted by the BitTorrent (Mainline), uTorrent, rTorrent, Transmission, KTorrent and BitComet clients.

Some considerations about BitTorrent:

- Peers interested for the same torrent form an independent mesh overlay network where each peer work together for a common target to complete the file download as fast as possible.
- One file can be registered in more than one tracker.
- When a user finishes downloading a file he can stop sharing or continue uploading.
- Trackers can store information of the data transferred by users.
- In some private trackers users must have a ratio of data_uploaded/data_downloaded higher than one threshold (for example, 0.8 or 1).

According to [11], BitTorrent has the highest percentage of Internet use. More than 50% of traffic generated by P2P networks is produced for BitTorrent network. The success of BitTorrent is because users focuses only in download a few files. Total users' band width has not to be split in many files, as happens in eDonkey2000.

3 Traffic Localization

3.1 General idea of Traffic Localization

Before the explosion of P2P networks, in the beginnings of 2000s, the mean traffic generated by a regular user was asymmetric. The reason is that all the services had a client-server architecture and its main characteristic is the asymmetry of the protocol (server uploads and user downloads). On the other hand, one of the main characteristics of P2P traffic is the symmetry. Moreover, depending on the client used to connect to one P2P network, users that share more content are rewarded with more credits (system credits used in some networks for example eMule, BitTorrent, etc.). The mechanism of this system credits is easy. The more you upload, the more priority you have in other users queues. The implementation of this system was due to a problem called free-riding.

Free-riding is a well-known problem of most decentralized networking systems, where users are expected to contribute some resources to make the system operational. There are a number of reasons behind the free-riding problem. First of all, it is important to note that free-riding is not usually performed to intentionally disrupt operation of the system. Free-riding users prefer the system to be fully operational as they still get the service from it. Thus, one part of the problem is that casual users do not understand that limiting upload traffic may decrease the quality level of the service they receive and the decision about limiting outgoing traffic is related to the human nature (get more than you give). Another reason is related to that group of people who are perfectly aware of what they do. Although upload traffic is not charged separately by ISPs, huge amount of it may serve as an indicator of extensive usage of P2P applications. These users intentionally limit the uploading rate of clients to hide their activities. Finally, free-riding is a natural choice for those who have very limited upload rates (most common connections of ADSL).

As we commented in chapter 2, every user is acting at the same time as a user and as a server. So now, with P2P networks, users wants to take advantage of all their bandwidth to upload as much as he can in order to be rewarded with more credits and improve their experience on this kind of networks. So with this new mentality it appears an additional problem for ISP. There is a new-extra traffic generated by P2P networks. Furthermore, most of this new traffic goes through third networks. Therefore, P2P systems generate an abundance of inter-ISP traffic.

As we know, Internet is a network that is made up of multiple networks. Each network is property of one company, for example Sonera, Telefonica, Vodafone, AT&T, etc. If we want to connect computers that are in different networks we have to be able to connect to this foreign networks. So ISP have agreements with other ISP or third companies to arrive to this networks. With the introduction of P2P this agreement changed and is still changing. Now, the behavior of the users is different. The amount of traffic that users consumes from other networks is increasing. In other words, users pass to be consumers of data to be producers of data.

For ISP, this new interaction between users of different networks is a problem. Now the agreements that they have reach is changing and becoming less favorable to their interests. So, because of the P2P they have to renegotiate new contracts and having less profits.

The root of the problem of the new inter-ISP traffic generated is that users choose their peers randomly. So, if we want to download a file that is shared worldwide and there are peers around the world we are not going to make any difference between a peer that is in our same network or another one that is far away from us.

Figure 3 is an example of how users from different countries interact each others to transfer files. In this example we have printed one user from Spain transferring information between random users distributed in Europe. In this example, this user does not interact with users from his own country.

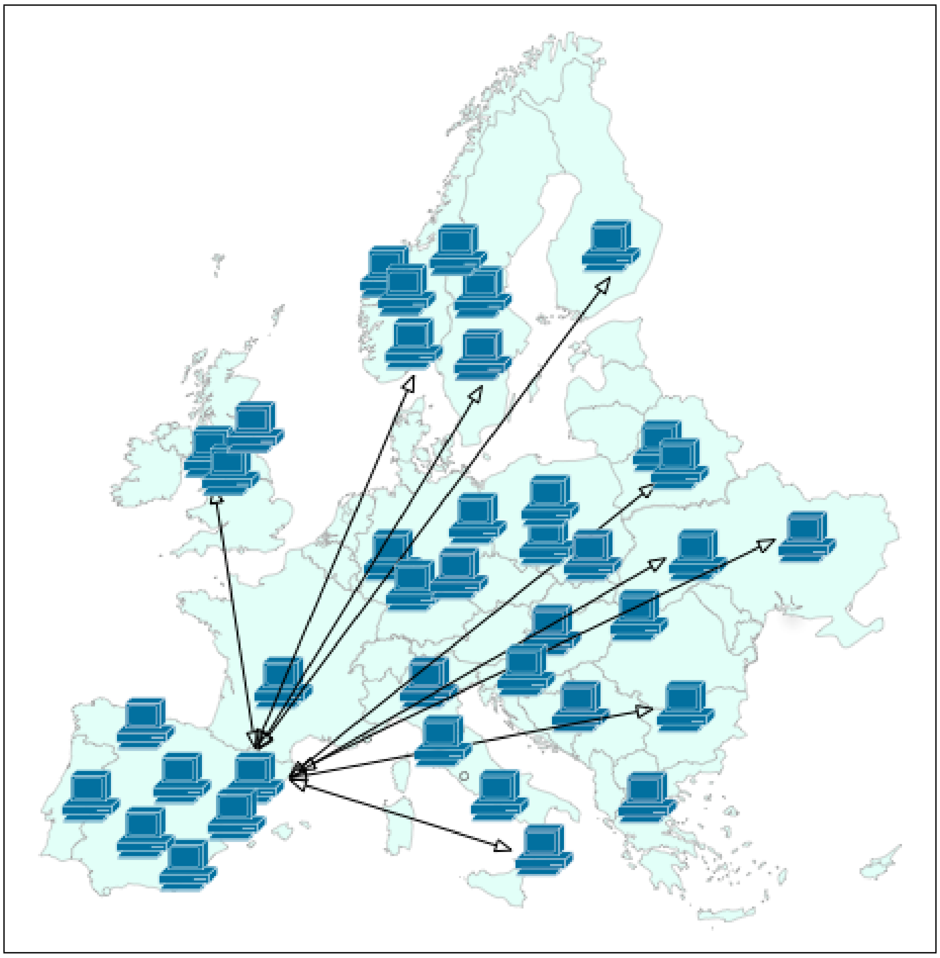

Figure 3: Example of random peers selection.

There are some ISP that are adopting “radical” solutions to minimize this increase of P2P in their infrastructure. One solution is to limit or even ban traffic that are generated by P2P. Some companies in Spain (ONO) have filtering process to determinate which traffic is generated by P2P applications and then they limit the bandwidth of this kind of applications. Some studies made from internet webpages and forums [2] indicates that this company reduces P2P traffic up to 15 Kbps. It means that, if one user connection has a 50 Mbit/s connecgtion he can browse internet pages, watch YouTube videos, downloading a file from a FTP server,etc. a 50 Mbit/s. In P2P networks this speed is going to decrease drastically (15 Kbps). The main reason that companies argue for take this measures is to offer rest of their clients a best service.

Also, there are another most extended prohibition of P2P traffic. Many mobile phone companies deny access to P2P network to their clients. This prohibition is usually applied in form of extra fees charges. The main reason is one popular service that P2P can offer, VoIP. If users can use internet fee for making calls instead of use traditional system, benefits for companies will decrease. For this reason, exists one principle called Network Neutrality. Network neutrality is a principle proposed for residential broadband networks and potentially applicable to all communication networks. It describes what should be the manipulation of the traffic through this networks. A neutral network is one that is free of restrictions on the kinds of equipment that can be used and modes of communication allowed, which does not restrict content, sites and platforms and where communication is not unreasonably degraded by other communication.

The increase in traffic due to P2P networks and especially file sharing has forced telecommunications companies to change their internet rates. Most internet rates include an unlimited amount of data. Another solution adopted by operators is to offer a maximum data per monthly/week. For example, 1GB/month connections. Extra data consumed has to be paid.

There have been set out many solutions to this problem; for example use an oracle service that inform every user who is the optimum peer to connect with. Also we can use caching of P2P traffic, usage of gateway nodes and biased choice of peers.

The “oracle service” is a service that the ISP will provide to their users. The procedure is simple. When the user has to decide which peers is going to select to download a file he asks the oracle instead of select it randomly. The oracle is going to be responsible to select peers that are near the user and have the corresponding parts of the file. The concept of this service is easy but it has not been implemented for “political” or “legal” reasons, not technical. ISPs do not want to have any kind of responsibility for the content distributed in P2P systems. It may lead to legal issues with copyright holders (RIAA, SGAE,etc.).

Another possible solution is to catch P2P traffic or usage of gateway nodes [9]. This technique uses proxy servers. A proxy server is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource, available from a different server. The proxy server evaluates the request according to its filtering rules. Depending on the filtering rules proxy can catch the packet or redirect to the original destiny. This solution is also very simple and could save much traffic to the ISP because proxies are usually installed in the ISP networks. So, when we are requesting for a file that is in another network ant the proxy server catch it, we are saving this traffic. As happens with "oracle services", catch P2P traffic means that ISP has some active part in this networks.

The last purpose, biased choice of peers is the most studied. The main problem of the interAS problem is that in P2P systems we choose peers randomly. This techniques works on making this choice depending on some factors. This factors are numerous and can be combined with other techniques. One possible element to be taken into account when we are choosing a peer is the number of hoops from the user to the destiny. This data is easy to find out (for example, IP protocol incorporate field TTL (Time To Live) that measures the number of hoops that one packet is passing before arrives to the destiny. Also, another element that we can use to find out the information of one peer is the information provided by DNS (Domain Name System). There are more parameters and ideas to discover this information, for example [15].

All of this mechanisms has the same purpose. To divide one network in small clusters trying to minimize the traffic that clusters interexchange. This technique is called Traffic Localization. Clusters can be divided by many parameters, for example, logical distance (number of hoop from one user to other), physical distance (for example, group users from the same city)[15], users from same company, etc. Also, it could be a very good idea to mix some of this techniques to find the optimum. Traffic localization is an idea of sectorization, there is no definitive way or parameter to classify one cluster or network.

We can say that the way of make clusters, number of interconnections, relation of users/clusters and many other factor to take into account will decide the success of this techniques. For example, the behavior of the network could be totally different if we make too much clusters or/and connections.

In figure 4 we can see an example of a network with clusters. Clusters are represented with clouds. In every cloud there are a group of users selected by some filter: distance, network, etc. Then, this cluster is going to be connected to other clusters thought only some users. Users from same cluster can connected each other.

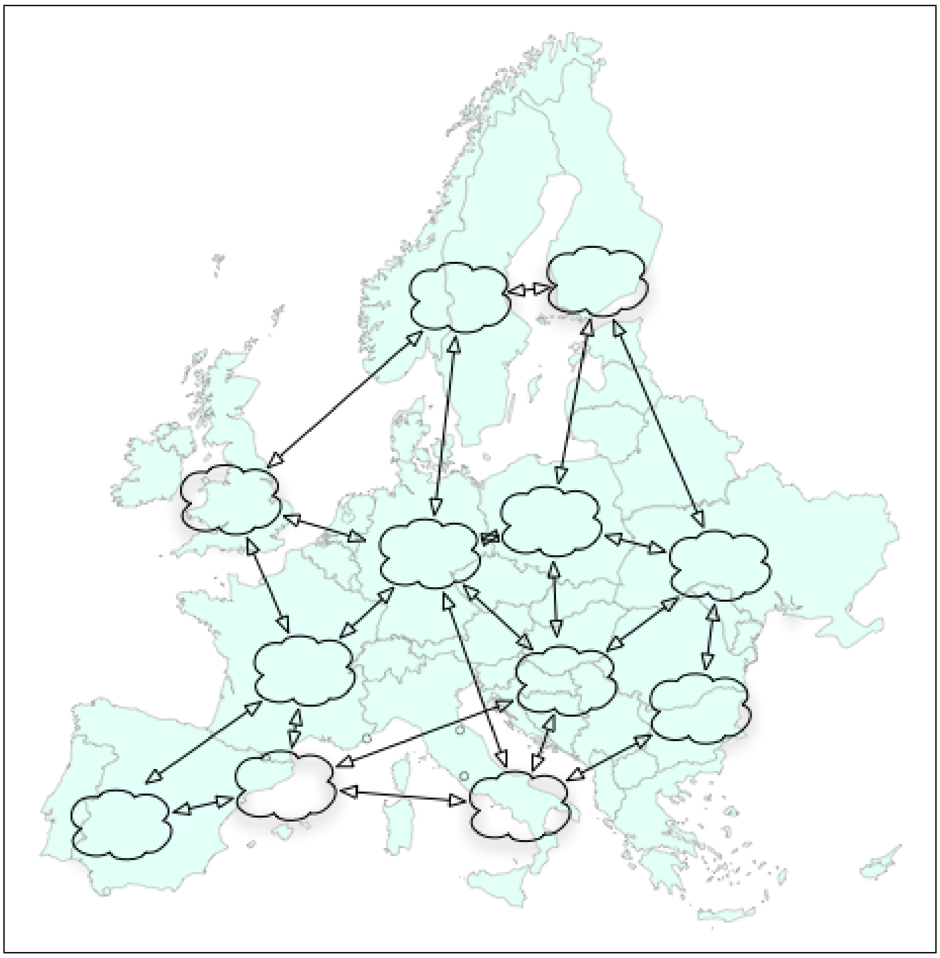

Figure 4: Example of Traffic Localization

Traffic Localization is a good idea to save resources in networks, but not all the networks/architecture are available to this technique. In some scenarios like web browsing or mail storage, it's very difficult to redistribute the traffic because the main servers are localized in one specific network and place. One example could be Gmail web server. Google has his own data centers distributed around the world. Our ISP has an agreement with a third company to be able to arrive to Google networks.

Exchange traffic is very important not only for ISP. In fact, some big companies (for example Google) that offer services that demands huge band width (like YouTube, Gmail, Google.com, etc.) have constructed their own fiber optic network. It connects all their datacenters around the world. ISP needs to arrive to Google's datacenters so they have to cross Google networks. ISP have to reach an agreement with Google or other companies to arrive to this datacenters. With this example, we can see how important is bandwidth for ISP and companies. In this case, Google built and is still building a hole infrastructure between their datacenters in order to save money or in some cases to earn money from ISP agreements. On february of 2011 responsible for China Mobile, AT & T, Vodafone, America Movil and Telefonica claimed from the Mobile World Congress in Barcelona by the end of the regulatory barriers that discriminate against companies telcos against other digital actors, creating an open and the allocation of network costs, only measures, in his opinion, capable of boosting demand for innovative services and applications required by the new digital world.

In P2P networks, we do not have only one possible destination for download a file, we have as many possibilities as peers are in the network. This is the most important advantage of P2P networks. The amount of resources of the networks is proportional to the number of users that form it.

3.2 Positive effects

There are numerous studies studying the positive effects of traffic localization [4, 5, 6, 8, 9, 15, 22, 24]. There are many documented positive effects of Traffic Localization.

The most important benefit of traffic localization is the reduction of intra and inter AS traffic. In [4] there is an example of reduction of inter AS traffic using an efficient traffic-exchange localization algorithm. In this example, peers tries to obtain higher portion of data from nearby partners than remote ones. There are more studies that show how using localization techniques decrease the average time of downloading a file, for example [5] and [9].

3.3 Negative effects

As the opposite that happens with the positive effects of traffic localization, there is not an abundance of studies and information about the negative effects of Traffic Localization.

One of the well-known problems of Traffic Localization is the increase of the average time. When a network is clusterized we are also dividing the total bandwidth that peers are going to be able to reach.

Also, it is important to note that the degree of effectiveness is somewhat limited. For example, language barriers still exist between countries and the media content that is popular in, say, Germany, may not receive a lot of attention elsewhere naturally leading to countrywide localized traffic. Although this observation is valid for a significant part of P2P streaming content, there is a lot of live content that is popular around the world irrespective of the language of audio information. For example, the latter is true for major sporting events such as FIFA World Cup or Olympic games. Moreover, even when a certain content is popular within a single country it still may lead to high amount of inter-ISP traffic, especially, when mid- and small-size ISPs are concerned.

Another negative effect is that some localization schemes may lead to the clustering of peers in P2P systems. In this case even inherently robust systems, such as those based on mesh overlays, become vulnerable to service interruptions as a result of the loss of connectivity between clusters.

4 Simulation And Results

In this section we are going to start our work. We are going to choose, install, modify and configure one BitTorrent simulator. We choose *Ubuntu 10.10* as operating system with all the updates available. Compiler and GUI were the default included in *Ubuntu*.

4.1 Simulators

In order to be able to simulate a network with traffic localized we need to choose very careful a simulator. In our research we took into consideration the following elements:

1. Flexibility: We have to be ables to modify the code of any part of the modules, connections, routing tables, interface tables, etc.
2. Scalability: High number of possible nodes that the simulator can manage.
3. Efficiency: Skill for manage via scripts as many simulations as we want.
4. BitTorrent model protocol available.
5. Update: The project of the simulator needs to be actualized.
6. Interactive GUI: Skill for manage and see graphically our simulations.
7. Language used has to be familiar (C,C++, java, etc.).
8. Information available from tutorials and community forums.
9. Open Source code.

We founded many simulators with this properties. Here is a list of possibles simulators that we founded and we take into consideration for our work: OverSim [13], PeerSim [19], P2PReal [20], GPS [27], NetSim, ns2/ns3, Microsoft BitTorrent Simulator [18].

Every simulator has his own advantages and disadvantages. All of them have possibles implementations of the BitTorrent protocol. Some of them were written for simulate only BitTorrent network/protocol. Other ones are "just" network simulators but exists extensions or plugins to simulate other protocols (BitTorrent in our case).

There is no "perfect" simulator with only advantages and no disadvantages. For example, NS-2 has low scalability but exist many information about it, GPS has no BitTorrent model but it's written on Java, Microsoft BitTorrent simulator has no flexibility, PeerSim is out of date but it has a BitTorrent implementation, etc.

After discussing which simulator will be the most efficient for our work, we decided to use the simulator OverSim. The main reasons for use this simulator are that is based on OMNeT++, has a good flexibility and scalability. It is written in C++ and there are an extend number of FAQ's and forums with documentation, problems and help. Also, exits a BitTorrent model [16].

4.2 OverSim

4.2.1 Version

Once we have chosen the simulator that we are going to use, next step was to choose the correct version. OverSim is an open-source overlay and peer-to-peer network simulation framework for the *OMNeT++* simulation environment. We chose:

1. OverSim version 20080416
2. INET version 20061020
3. BitTorrent model 1.1

Now a days *OverSim* is in *OverSim-20101103* version that uses the *INET-20101019* version. We had to choose *OverSim*-20080416 with Internet patch *INET* -20061020 because the BitTorrent model existing was written for these software's versions. People from [10] are working in the new model of BitTorrent. This model is, among many other things, going to work with new *OMNeT++ 4.x* version.

4.2.2 Installation

OverSim version 20080416 was released in 2008. In 2008 the Linux compiler was GCC 3.x. So, in order to be able to compile it, we have to make some modifications to headers' files in OMNeT++, OverSim and INET source code. Current version of OverSim, INET framework and OMNeT++ do not need any kind of modification.

Also, we have to add manually some libraries, for example *iostream.h*. This libraries are out of date and are not included in modern Linux distributions.

After been successful installing OMNet++, INET framework and OverSim we have to start implementing BitTorrent 1.1 model into our OverSim simulator. In the process of integration BitTorrent model we had to take care of the following issues:

1. There has been some modifications on the structure of INET framework.
2. Some default BitTorrent parameters are missing [21].
3. Some steps of *dynamicBTNodeDeployment.cc* needs some extra modifications [21].

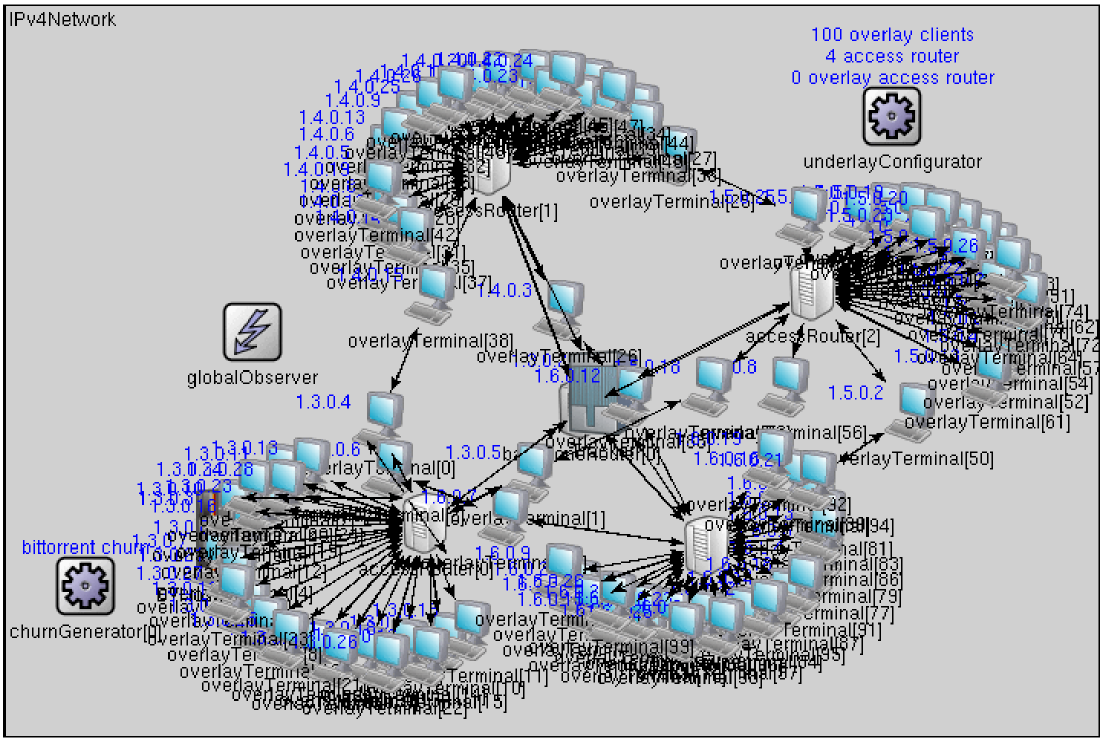

Figure 5: Example of OverSim with GUI

Figure 5 is an example of OverSim working with GUI. In this example we can appreciate 4 routers, one tracker and 100 of nodes.

4.2.3 Structure

OverSim is based on OMNeT++ and INET framework. So, we need to have both programs installed to run OverSim. We have created a folder called *sim* in our home directory. In this folder there are the three programs. So we have:

1. */sim/omnetpp*
2. */sim/INET*
3. */sim/OverSim*

Our simulator will be in */sim/OverSim/*. In this folder will be save all our parameters and all our results.

4.2.4 Configuration and Parameters

Once our simulator is installed, we have to take care to review some parameters of the BitTorrent model. For example, parameters saved in *channels.ned* file. A *channel* is the description of peers' connection. The default connections preconfigured in OverSim are out of date. During the development of the BitTorrent model (2005 to 2008), 56Kbit connection was a regular connection. Now a days, such kind of connections are almost nonexistent. Therefore, we decide to update them into more modern connections. We have used the same connection for all the users (MT) although the simulator is prepared for choose one aleatory channel.

Channel list reviewed:

1. Fiberline: datarate 10*100000000; delay 1ms.
2. tracker: datarate 10*100000000; delay 0ms.
3. MT: datarate 5000000; delay 50ms.
4. dsl: datarate 1*1000000; delay 50ms.
5. ethernetline: datarate 1*10000000; delay 10ms.

tracker will be the channel for the tracker. *Fiberline* will be the connection of the seeders in Ring Topology. And *MT* will be the connection of initial and regular nodes and seeder in Mesh Topology.

BitTorrent module needs parameters to run simulations. Most of them are predefined but many other are not. The default parameters file are in */sim/OverSim/ Simulations/defaultBitTorrent.ini*. Among other things, we have to select which channels are going to use our tracker and our seeder, the function and value of the distribution of arriving time of peers, number of peers, etc. In the following table are are a summary of those parameters:

General Parameters	**Values**
BTHostseeder.channelType	MT, fiberline
decreasingExponentialArrivalRate	0.00166
accessRouterNum	Number of Routers
backboneRouterNum	Number of BR
globalObserver	BTStatistics

Parameters tracker	**Values**
tracker.channelType	tracker
trackerClient.peerPort	6881
sessionTimeout	600
announceInterval	300
cleanupInterval	600

Parameters seeder	**Values**
BTHostseeder.channelType	MT, fiberline
peerwire.file size	1Mb
peerwire.piece size	256
peerwire.block size	16
peerwire.downloaders	50
peerWire.timeToSeed	2000seconds

For being able to construct our topology we had to modify some functions and add some new parameters. Here is a table with our new parameters:

New Parameters	**Values**
numFixedHosts	Number Fixed Nodes
numBTHosts	Number seeders
typeOFinterconnection	0,1,2

Before start simulating our clusterized network, first of all we have to determinate a formula that quantifies the range or the "quantity" of "Traffic Localization" that are in our network. We have used the following formula for Ring Topology:

$$\frac{N}{k} * \rho = n$$

Where:

N is the number of nodes in system.
k is the number of clusters.
rho is the coefficient of sectorization.
n is the numbers of connections between clusters.

According to the formula and for a set of nodes (N) of 400. We obtain the following values:

N of clusters	ro=0,1	ro =0,2	ro=0,3	ro=0,4
4	10	20	30	40
8	5	10	15	20
12	3	7	10	13
16	3	5	8	10

In Mesh Topology, where the number of interconnections is not proportional to the number of clusters, we are forced to use a constant number of interconnections. For example, a network with 400 nodes, ro = 0,4, and 16 clusters has 10 interconnections in Ring Topology and 160 in Mesh Topology.

So, in Mesh Topology we were forced to use a constant number of interconnections 1, 4, 7, 10, 13 and 16 interconnections independently of the number of clusters and the number of nodes in the network.

Once our simulator is working as a regular BitTorrent system, we must now construct the clusters of the network.

4.2.5 Modification

The idea of the modification is to change the behavior of the network. We need that the structure of the P2P network behaves like clusters. So, we have to modify the following elements of the network:

1. Router: Each router of the networks will have access only to the nodes connected to the same network. But, routers will not be able to reach other routers. Therefore, in order to mimic this behavior, we have modified the routing table of every router. Here is an example of a routing table with 2 nodes connected:

Description	**Network destination**	**Netmask**	**Interface**
Loopback network	127.0.0.0	255.0.0.0	if0
tracker	1.3.0.2	255.255.255.255	if1
Node 1	1.3.0.3	255.255.255.255	if2
Node 2	1.3.0.4	255.255.255.255	if3
Node X	1.3.0.X	255.255.255.255	ifX
Default route	127.0.0.0	255.0.0.0	if0

Every user that gets connected to the router is going to create a new entry in the interface table and in the routing table.

2. Tracker: tracker is the common element for every node in our network. Seeders, initial nodes and regular nodes need to connect to the tracker to exchange information. As we just have seen, routers cannot reach other routers so we cannot put tracker in only one router (other elements of the network would not be able to reach the tracker). Our solution was to connect the tracker to every router in the network. The interface table of the tracker and routing table has one entry for every router of the network.

3. Seeder: In our simulations we only have one seeder notwithstanding we implemented the possibility to create as many seeder as we want. Seeders have the same category that regular nodes except for the creation time. It is created in $t=0$ *s* and it's connected to router_0 (1.3.0.1).

4. Initial or Fixed Nodes: Initial or Fixed nodes are nodes that form the topology of the network. Those nodes have 2 connections. First, it is connected to only one router. With this connection it is going to reach all the users of the same cluster. Second, it is connected directly to one user from other cluster. With this connection this kind of user are going to be able to exchange data with another node from other cluster, but only one of this router (not other ones from the same cluster).

   Here is an example of a routing table:

Description	**Network destination**	**Netmask**	**Interface**
Loopback network	127.0.0.0	255.0.0.0	if0
Router n	1.(3+n).0.1	255.255.255.0	if(1+n)
Peer.1	1.x.0.4	255.255.255.0	if2

5. Regular nodes: Regular nodes are basically the default nodes implemented in the basic BitTorrent model. They are created at random moment and connect to one router. In this router, there are a connection to the tracker and (excepting the router.0) other peers but not seeders.

For this purpose, we edited the files *IPv4 Underlay Configurator.cc* and *Access.cc* files. Those files are in */sim/OverSim/Underlay/IPv4Underlay* folder. We add some new functions, that help us to create our scenario:

- createNodeStatic: This function creates a node in t= 0s. Also, this node is saved in a vector of cModules called accessFix.
- connectandcreate: this function has parameters (cModule node, cModule router). This function creates an entry in routing table and interface table of the object node and object router. Then it connect gates, internal gates and queues. At the end of the function, both cModules are connected. This function can be used for connect router with nodes or nodes with nodes.

We also have edited some functions. For example function *initializeUnderlay*. In this function we start the process of creation of the fixed elements (tracker, seeder and fixed nodes), and we start to connect them forming our topology. In this point we added a new parameter, this parameter is called *typeOFinterconnection* and the parameter is located in *defaultBitTorrent.ini*. If the value is "0", we create a Ring Topology, if it's "1" we create a Mesh Topology with a propotional number of interconnections; if is "2", we create a Mesh Topology with constant number of interconnections.

In function *createBTNode* we change the default routing table of the routers into another ones that function as we have already explained. Also, we connect the tracker to every router in the network.

4.2.6 Basic operation

Parameters are saved in *omnetpp.ini*, *default.ini* and *defaultBitTorrent.ini*. Those files are in *sim/OverSim/Simulations* folder. In *default.ini* are networks' general parameters of the (OverSim default networks). In *defaultBitTorrent.ini* are default parameters that are constant in all the BitTorrent simulations. In *omnetpp.ini* are stored parameters that affect current simulations.

In *omnetpp.ini* file we can configure many different simulations with different parameters called "Runs". During our work and following the wiki of Omnet++ and OverSim we have implement a script for run automatically all our simulations.

After the configuration and testing of our network it is very recommended to simulate without a graphic user interface (GUI). According to our results, simulations without a GUI save 4 times less time simulating. It can be done easily with the options: *-uCmdenv -cChordLarge* in the command line.

Also, we have disable the creation of file *omnetpp.vec*. In this file is saved all the movements that happens in our network; what time spend packet to arrive to destiny, ping times, etc. It can be done easily adding the parameter ***.enabled = no* in section *[OutVectors]* of parameters files..

In our work, we only need the time that spend one user to download the file. Depending on the topology that we are working on, we can also be interested in identify the time spent for users from one specific cluster. So, moreover of time we also save the IP of the node that has finished downloading the file. To do this, we modified the function *updateBitField* of the file *BTPeerWireBase.cc*. This file is in *sim/INET/Applications/BitTorrent/* folder. In this function we open a file called *downloadTIMES.txt* and we store our information there.

Once everything is ready we start our simulation using our script. Our script is located in */sim/OverSim/* folder and is called *run.sh*. The script will run all the simulations of *omnetpp.ini* file and the results will be saved in *downloadTIMES.txt* file in */sim/OverSim/* folder.

4.2.7 Processing Results

Depending on the line commented in the function *updateBitField* of the file */sim/INET/BitTorrent/BTPeerWireBase.cc* our results will be saved in different formats. Here is an example of the results when we do not want to save the IP-cluster that has finished.

RESULTS:
Time.node_1,Time.node_2,Time.node_3,...

When we do not want to make any difference between times of clusters our total average time is directly the average of the line. We used Matlab for do this kind of operations. On the other hand, when we are interested to know the average time of only one cluster the *downloadTIMEs.txt* has this format:

RESULTS:
IP_1 Time.node_1
IP_2 Time.node_2
IP_3 Time.node_3
......
IP_n Time.node_n

So now, if we want to do the average time we have to pick only those times that are from one specific cluster. For identify users from one cluster we can use their IP. Second octet of IP address is the cluster identifier. All nodes have IP: 1.#.0.X. Where # is the number of the cluster and the value is [3,3+number of clusters].

If we are simulating networks of 1200 nodes, this task will be too difficult and long. To facilitate this task we have created some programs that will split the *downloadTIMES.txt* file, select times according to the cluster and save the average time in a *txt* file. The name of those files are:

1. *split*: Open *downloadTIMES.txt* file and create as many files with the results of the different simulations.
2. *run*: Open files created by *split*, then for each file it takes all the times of one cluster and makes the average. Results are saved in another file *results.txt*.
3. *run2*: This script transform *results.txt* to excel format. New file is called *results_excel.txt*.

Run, *run2* and *split* are executed consecutively and automatically for a script called *run.sh*. In this script we have to enter the name of the files that the function *split* is going to create (one per simulation done).

Also, we have made another script called *mean*. It makes the mean of the simulation times when we are not saving the IP address in *downloadTIMES.txt* file.

4.2.8 Topologies

Once we have created our network (tracker, seeder, and fixed nodes); we have to connect them in order to form one cluster topology. We are going to call cluster to all nodes connected to one router. So, if we have n-routers, we are going to have n-clusters. We need to connect this clusters forming a topology. In our work we have simulated two topologies. Ring Topology and Mesh Topology.

Also, we have to connect and create the rest of the peers that are not part of the main topology. All of this nodes are created in *IPv4UnderlayConfigurator.cc*. This file is in */sim/OveSim/Unverlay/IPv4Underlay* folder.

In Ring Topology, nodes that do not conform the topology are created automatically. Nodes appear at a random time. Nodes are created in *BitTorrentChurn.cc*. This file is in */sim/OverSim/Common* folder.

In Mesh Topology, all the nodes are initially created at t=0s. Nodes are created in *IPv4UnderlayConfigurator.cc*.

Ring Topology

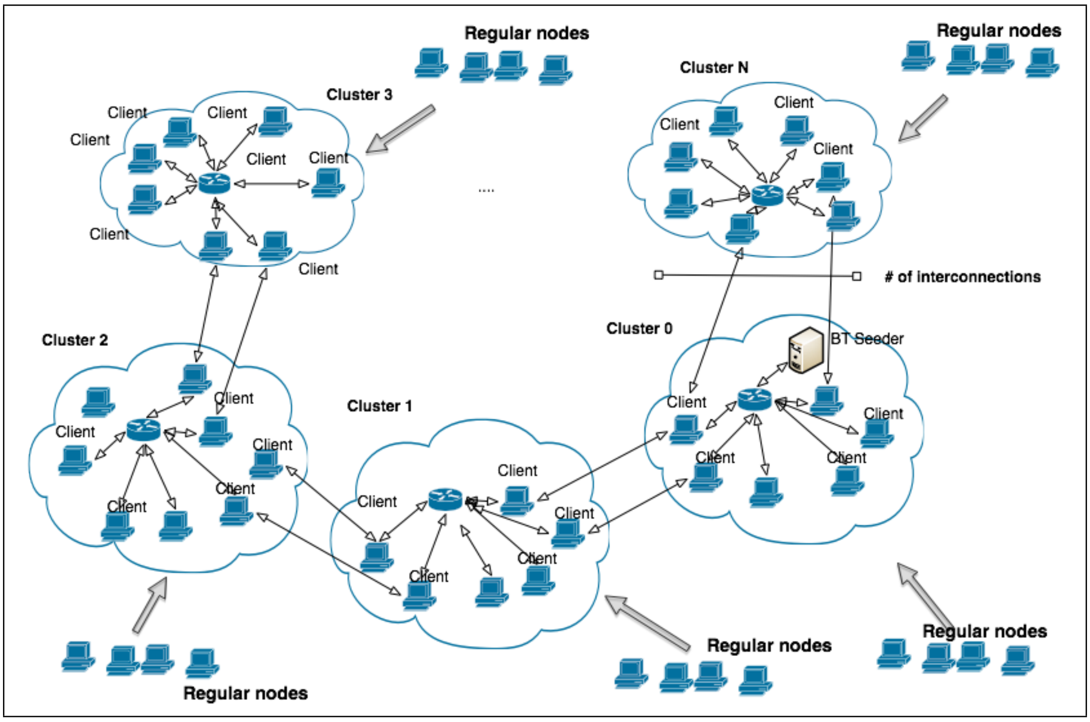

Figure 6: Ring Topology

Ring Topology has the less number of interconnections between clusters, we only need n connections between n-clusters. Figure 6 is an example of our topology. We can appreciate how clusters are connected with 2 connections between clusters. Also, as happens in our simulator, regular nodes arrives at one cluster before the initial time (t=0s).

Using rho parameter, here is an example of the number of interconnections between clusters, for 600 users, 6 clusters and ro=0,1:

Cluster	**0**	**1**	**2**	**3**	**4**	**5**	**TOTAL**
0	ALL	10	0	0	0	10	20
1	10	ALL	10	0	0	0	20
2	0	10	ALL	10	0	0	20
3	0	0	10	ALL	10	0	20
4	0	0	0	10	ALL	10	20
5	10	0	0	0	10	ALL	20

In this topology, if we increase the number of clusters, the number of interconnections increases proportionally to the number of cluster. If the number of nodes is N*2, the number of interconnections increases n*2.

In our work, we configured the simulator to create initially the tracker, then seeder and then *n* fixed nodes equally distributed in the k clusters. Fixed nodes are also created at t = 0s and they are connected to the router of their cluster and directly to other nodes of other clusters.

The rest of the users arrives at a random cluster. The distribution of clusters is given by the function *uniform [0,N-1]* where N is the total number of routers. The arrival time is given by the expression of the parameters, in our simulation: *exponential(0,00016)*.

Here is the number of interconnections for 4 and 6 clusters, 600 nodes and parameter rho = 0.1, 0.2, 0.3 and 0.4.

600 nodes	**0,1**	**0,2**	**0,3**	**0,4**
4 clusters	15	30	45	60
6 clusters	10	20	30	40

Here is the number of interconnections for 4 and 6 clusters, 1200 nodes and parameter rho = 0.1, 0.2, 0.3 and 0.4.

1200 nodes	**0,1**	**0,2**	**0,3**	**0,4**
4 clusters	30	60	90	120
6 clusters	20	40	60	80

Mesh Topology

Figure 7 is an example of our simulated Mesh Topology. We can say that it is a “full” Mesh Topology at cluster level. Every cluster is connected to all others clusters but not every peer of the cluster is connected to users of other clusters.

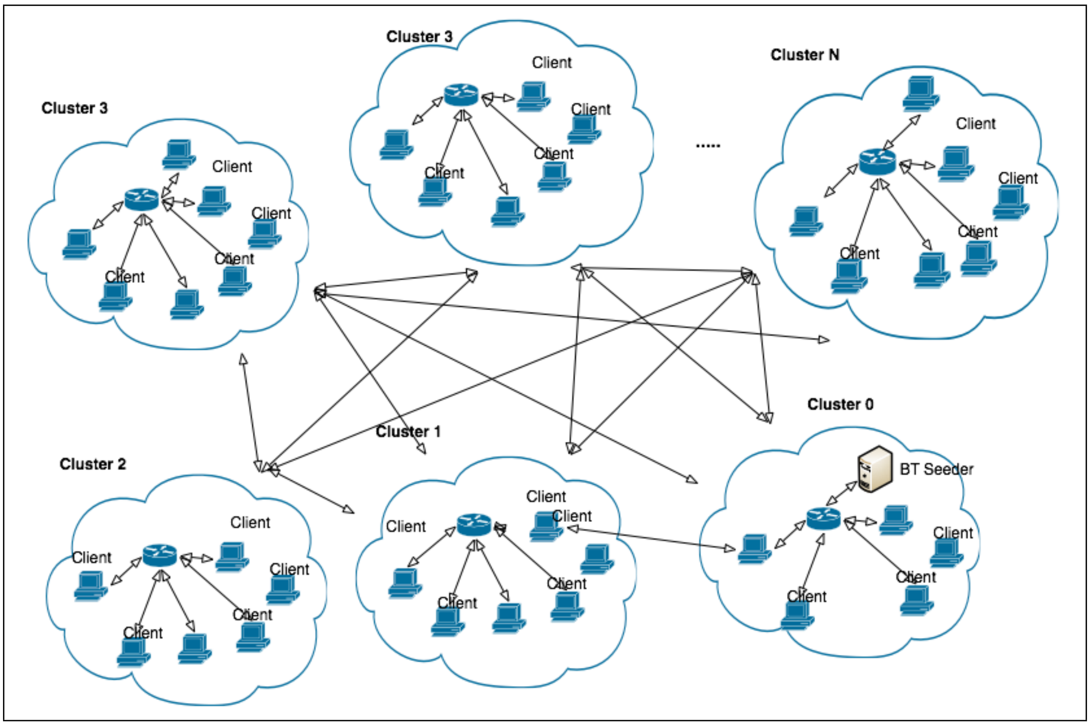

Figure 7: Mesh Topology

Mesh Topology has the maximum number of interconnections between clusters. The number of total interconnections between clusters is:

$$n = N * \frac{N-1}{2}$$

Where:
N is the number of nodes.
n is the numbers of connections between clusters.

In our simulations we have a number “N” of fixed connections between cluster 1, 4, 7, 10, 13 and16. This measure was taken because if we use the same Ring Topology parameter, the number of interconnections increases too much and the simulator was not able to work properly. Here is an example of the number of interconnections for a network with 4 clusters and 7 interconnections per clusters.

Cluster	**0**	**1**	**2**	**3**	**TOTAL**
0	ALL	7	7	7	21
1	7	ALL	7	7	21
2	7	7	ALL	7	21
3	7	7	7	ALL	21

All the peers of the same clusters are connected with each other thought the router. The first seven 7 nodes of every cluster are randomly connected to 7 nodes from other cluster. Then we repeat this operation since every cluster is connected to other clusters.

In Ring Topology not all the nodes are created in t = 0s (only tracker, seeder and the peers that form the topology). Now, in Mesh Topology all the users are created in t = 0 s. In this topology, if we increase the number of clusters, the number of interconnections increases more ((N-1)/2 times more).

4.3 Results

Once everything is ready, we start our simulations. We are going to simulate our network with both topologies (Ring Topology and Mesh Topology).

4.3.1 Results: Ring Topology

In Ring Topology there are different "categories" of clusters. We can split clusters according to the distance (number of hoops) that need to arrive to the cluster 0 (cluster that contains the seeder). For example, for 4 clusters we have 0, 1 and 2 -hoop distance. For 6 clusters we have 0, 1, 2 and 3 hoops distance. Figure 8 is a graphical example of distances in Ring Topology.

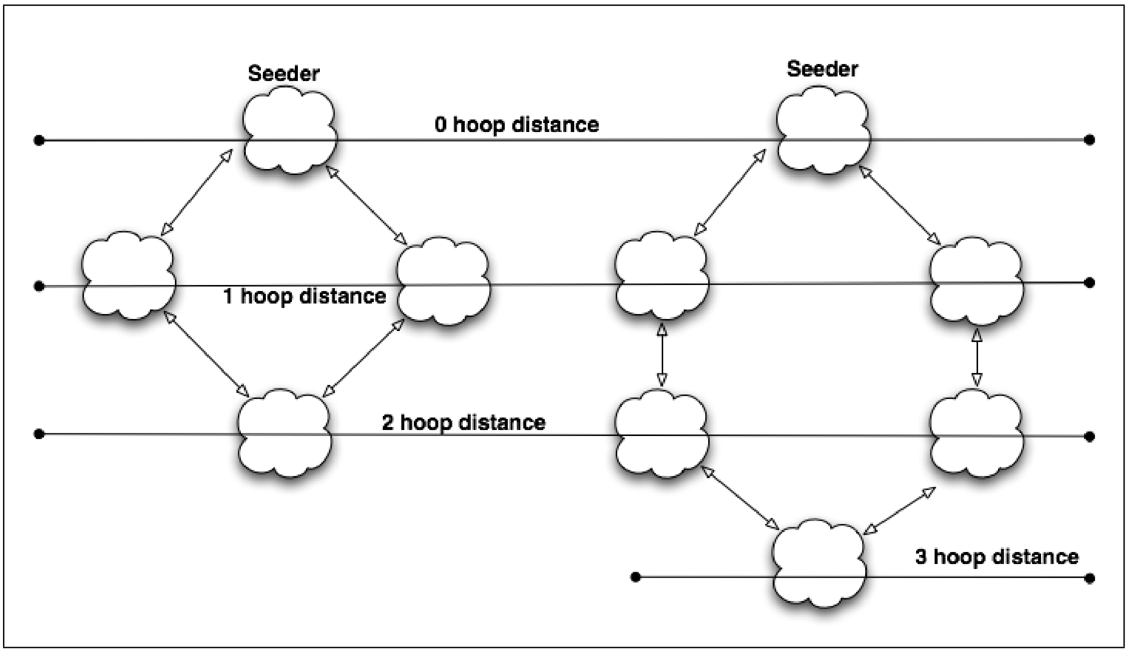

Figure 8: Ring Topology distances

In Ring Topology we are going to make our simulations separating the average time for every cluster. In our earlier simulations we made the total average time of all the users discriminating the number of hoops to the cluster 0. Our definitive results incorporate this new parameter.

Also, we have to take in mind that in our simulator and with Ring Topology, regular nodes does not arrive at the same time.

Experiment 1. Parameters:

- Topology: Ring.
- Number of nodes: 600 and 1200.
- File size: 10 Mb.
- Coefficient rho=0,1 0,2 0,3 0,4 .
- Clusters: 4.
- Initial nodes(t=0s):tracker, seeder and fixed nodes connected in Ring Topology.

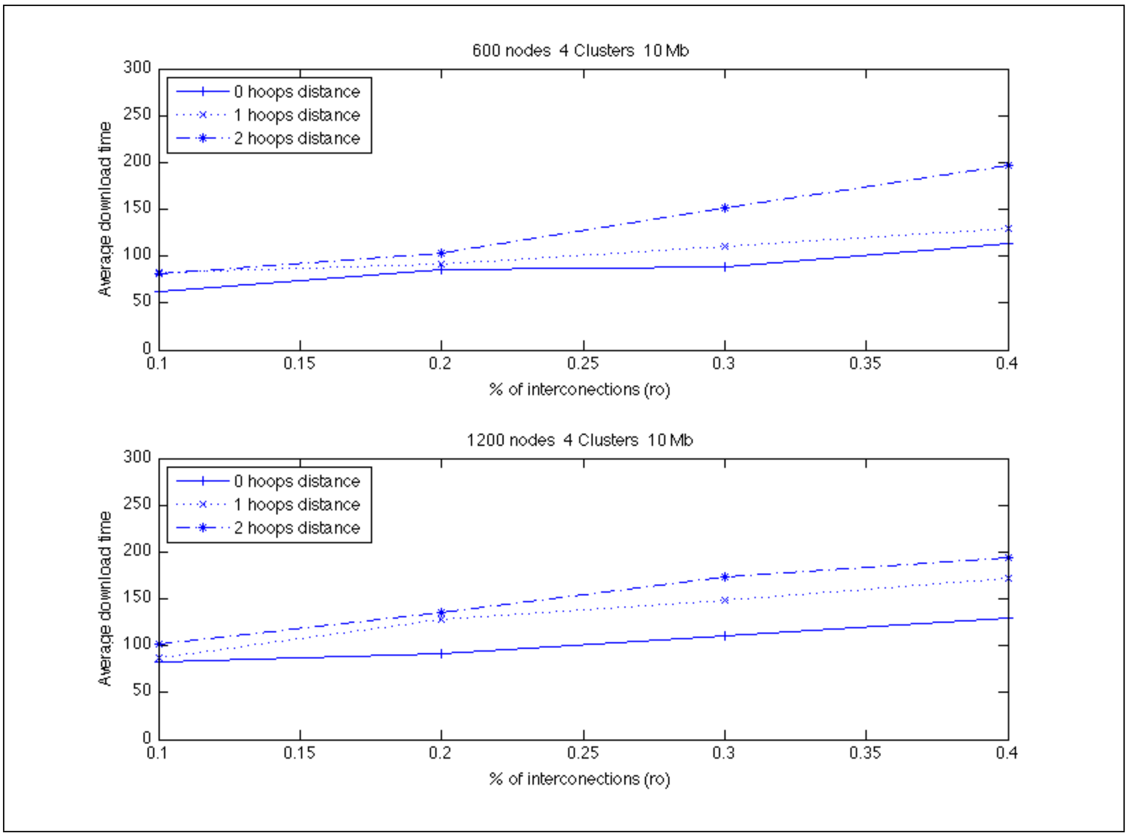

Figure 9: Results Ring Topology: 4 clusters 10 Mb

Experiment 2. Parameters:

- Topology: Ring.
- Number of nodes: 600 and 1200.
- File size: 20 Mb
- Coefficient rho=0,1 0,2 0,3 0,4.
- Clusters: 4.
- Initial nodes(t=0s):tracker, seeder and fixed nodes connected in Ring Topology.

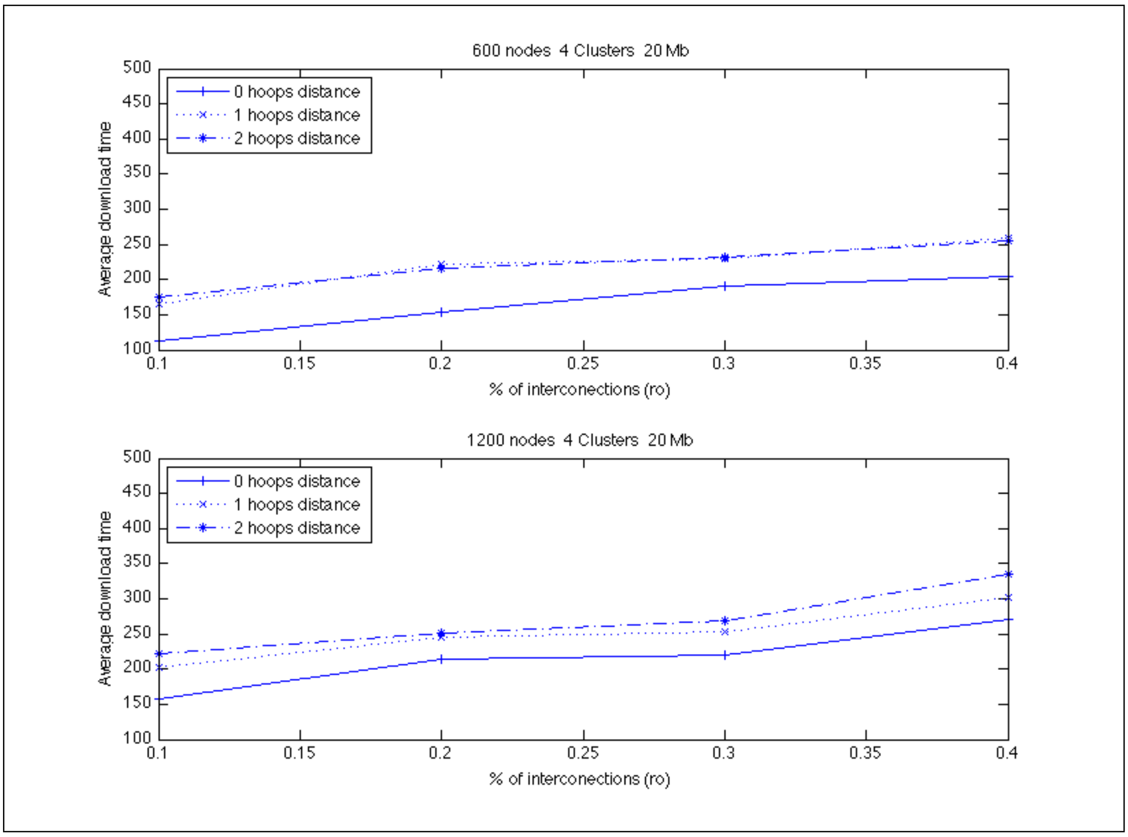

Figure 10: Results Ring Topology: 4 clusters 20 Mb

Experiment 3. Parameters:

- Topology: Ring
- Number of nodes: 600 and 1200
- File size: 10 Mb
- Coefficient rho=0,1 0,2 0,3 0,4
- Clusters: 6.
- Initial nodes(t=0s):tracker, seeder and fixed nodes connected in Ring Topology.

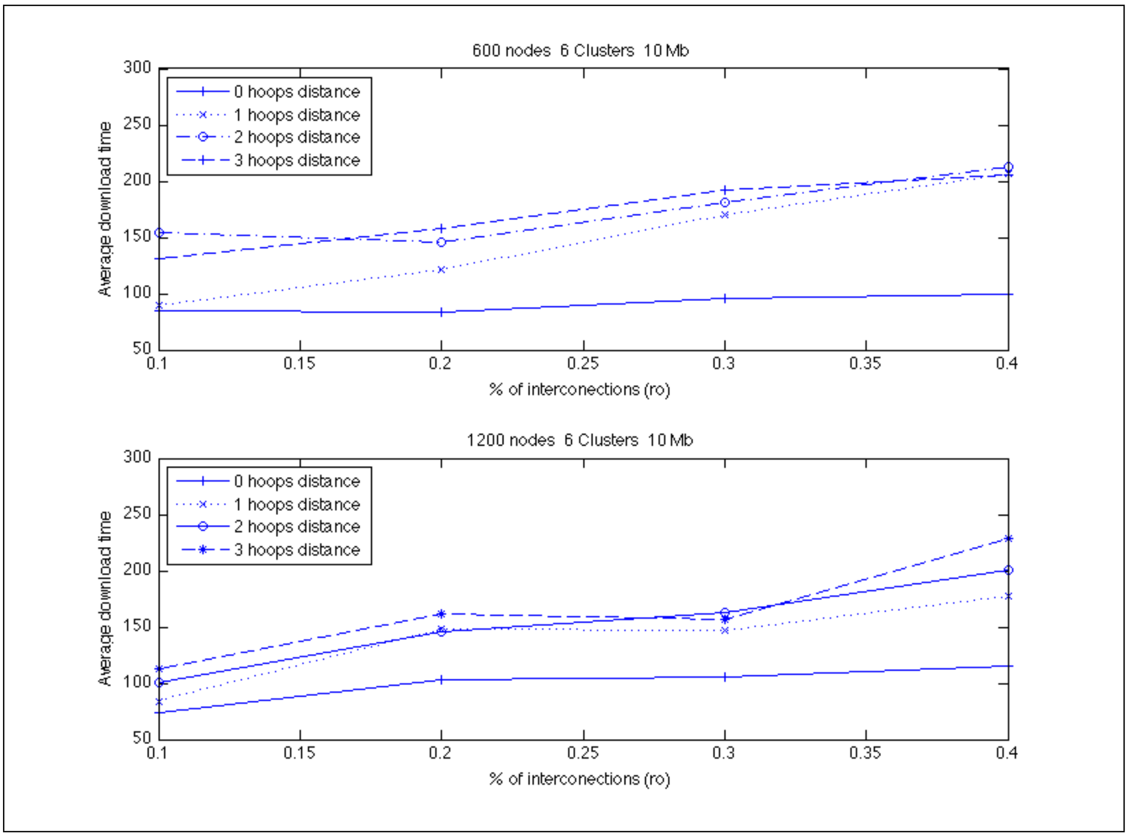

Figure 11: Results Ring Topology: 6 clusters 10 Mb

Experiment 4. Parameters:

- Topology: Ring.
- Number of nodes: 600 and 1200.
- File size: 20 Mb.
- Coefficient rho=0,1 0,2 0,3 0,4.
- Clusters: 6.
- Initial nodes(t=0s):tracker, seeder and fixed nodes connected in Ring Topology.

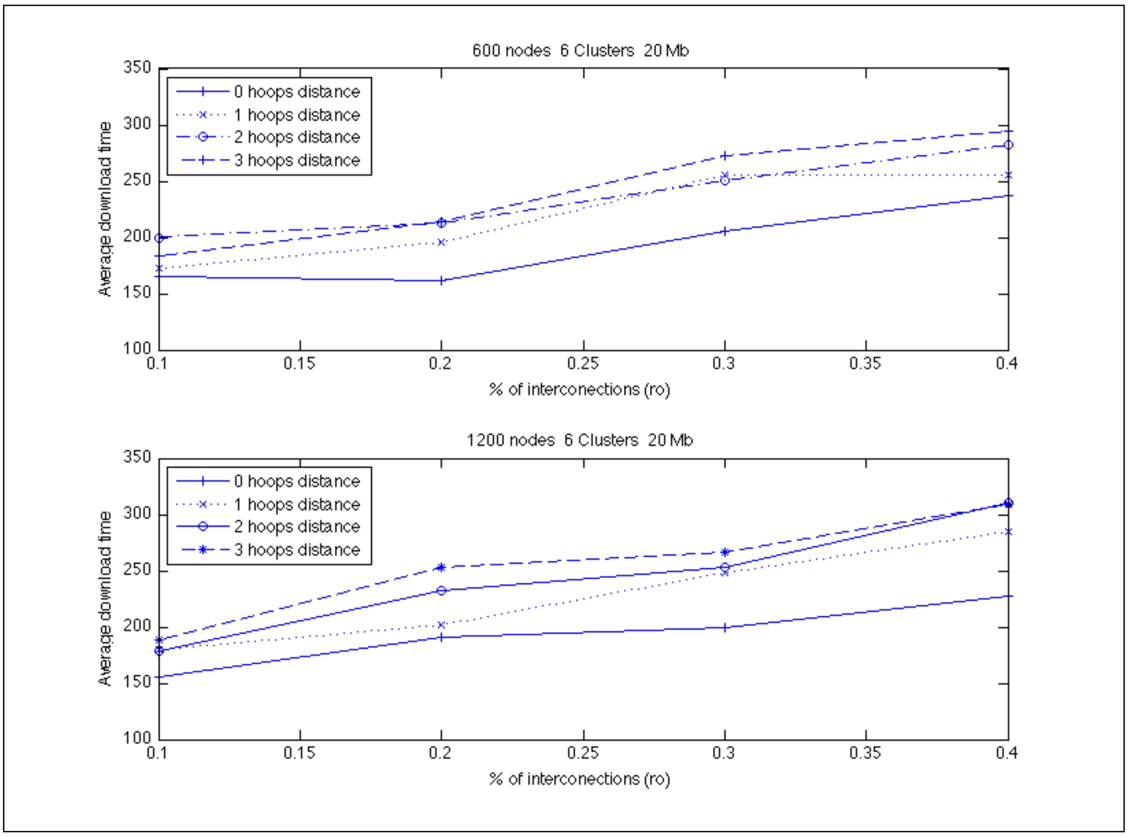

Figure 12: Results Ring Topology: 6 clusters 20 Mb

Y-axis represents time. X-axis represents rho parameter. Every line represents the average time spent for all the users contained in one cluster that is a 0, 1, 2,..N-hoop distance; to download a file.

As we can see on results represented in figures 9, 10, 11 and 12; if we increase the parameter rho, keeping number of clusters constant, the average time of one nodes increases. This is because when we increase rho, the total band width available in one cluster is going to be reduced and splited to the next 2 clusters. Increasing rho means more connections with next clusters, we are movoing resources from one cluster to the next one.

With Ring Topology, we have clusters with more priority. We can see this effect clearly in all the graphs. Cluster with more priority (for example cluster 0) are always on the bottom.

4.3.2 Results: Mesh Topology

In Mesh Topology there are no different "categories" of clusters. We cannot identify clusters according to the distance to the cluster_0 because every cluster is at distance 1. In Mesh Topology, for calculate the total average time of downloading a file we do not use the time of cluster 0, or 0-hoop distance information. The average time will be the average time of all the nodes of all the clusters except nodes from cluster 0. With this consideration our average and results will be more accurate.

Experiment 5. Parameters:

- Topology: Mesh.
- Number of nodes: 600.
- File size: 10 Mb.
- Number of interconnections between cluster: 1, 4, 7, 10 and 13.
- Clusters: 4, 6, 8, 10.

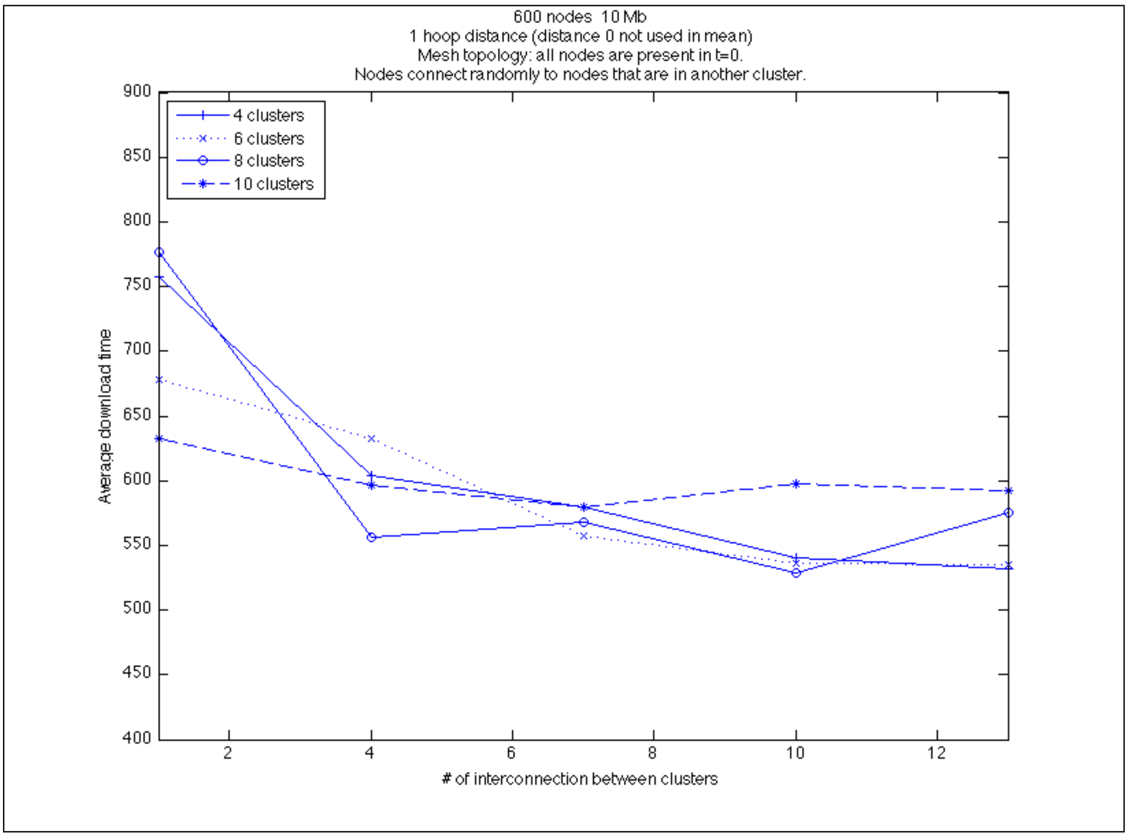

Figure 13: Results Mesh Topology: 600 users 10 Mb

Experiment 6. Parameters:

- Topology: Mesh.
- Number of nodes: 600.
- File size: 20 Mb.
- Number of interconnections between cluster: 1, 4, 7, 10 and 13.
- Clusters: 4, 6, 8, 10.
- Initial nodes (t=0 s): All.

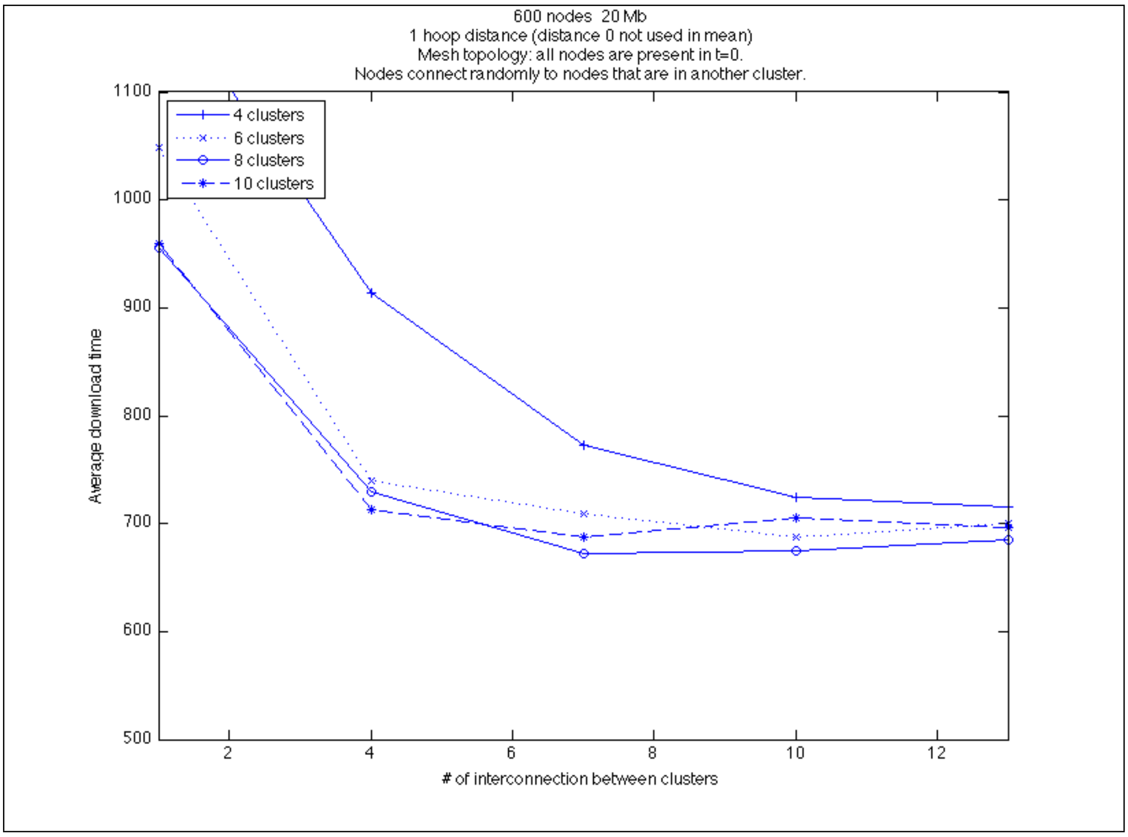

Figure 14: Results Mesh Topology: 600 users 20 Mb

Experiment 7. Parameters:

- Topology: Mesh.
- Number of nodes: 1200.
- File size: 10 Mb.
- Number of interconnections between cluster: 1, 4, 7, 10 and 13.
- Clusters: 4, 6, 8, 10.
- Initial nodes (t=0 s): All.

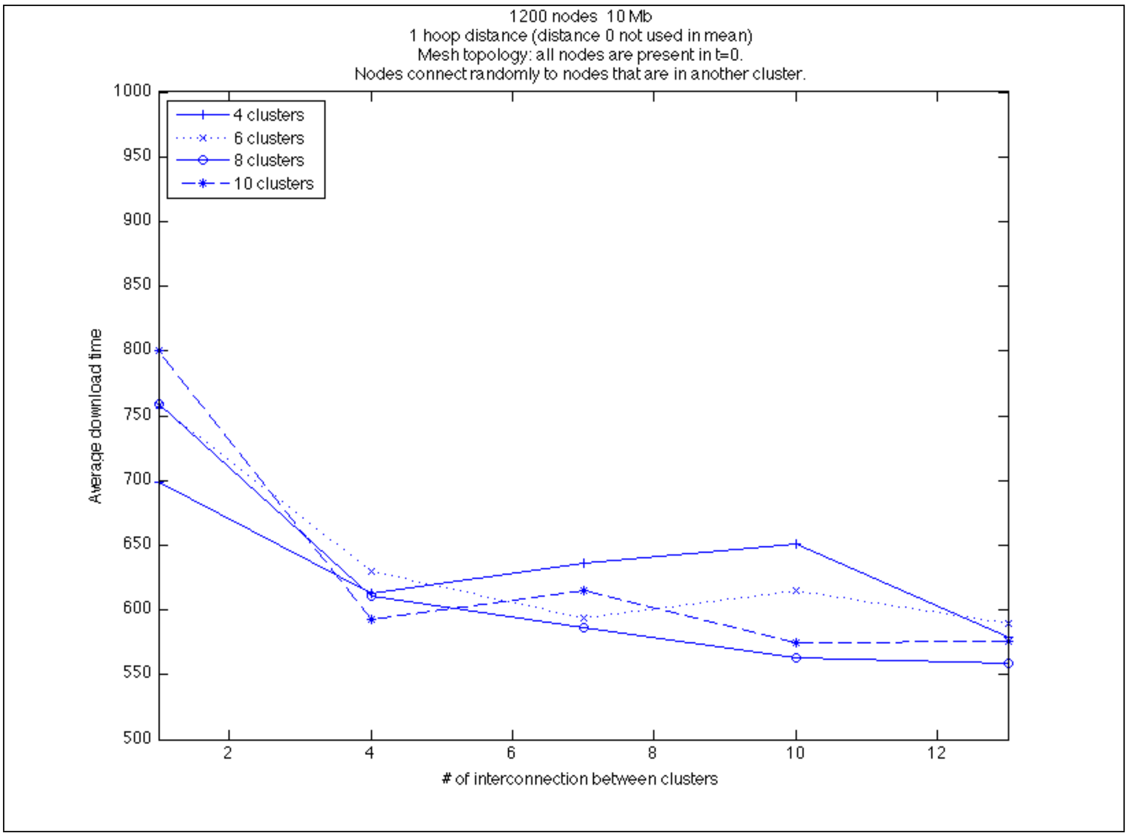

Figure 15: Results Mesh Topology: 1200 users 10 Mb

Experiment 8. Parameters:

- Topology: Mesh.
- Number of nodes: 1200.
- File size: 20 Mb.
- Number of interconnections between cluster: 1, 4, 7, 10 and 13.
- Clusters: 4, 6, 8, 10.
- Initial nodes (t=0 s): All.

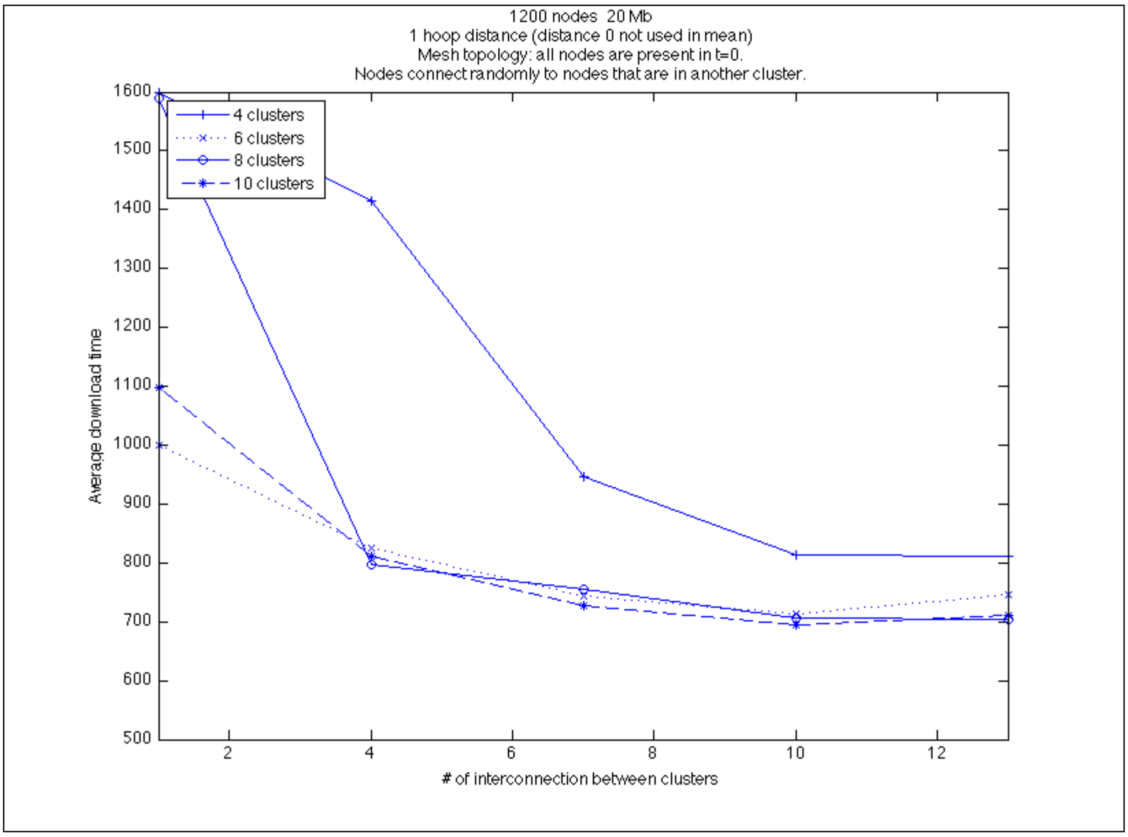

Figure 16: Results Mesh Topology: 1200 users 20 Mb

Y-axis represents time. X-axis represents the number of interconnections. Every line represents the average time spent for all the users of all the (N-1) clusters (except users from cluster 0); to download a file. Every line is for a different number of clusters (4, 6, 8, 10).

Figures 13, 14, 15 and 16 represents the results of our simulations. If we analyze the graphs obtained we can extract some information.

First of all, in all the simulation, if we increase the number of interconnections with a constant number of clusters (one single line), average time decreases. It means that if we have all the cluster "full" connected, increasing the interconnections decreases the time. This results is obvious because if we have more interconnections between cluster, file can be distributed more easily to all the clusters.

Another information that we can obtain is the behavior if we increase the number of clusters. As we can see in all the graphics, all the lines (average time for a fixed number of clusters) are similar. It means that when we have a *full* Mesh Topology the number of clusters is not a relevant parameter. As we told, every cluster is at 1-hoop distance increasing the number of clusters does not mean increase the distance to cluster 0.

5 Conclusions

The results that we have obtained with our several simulations have demonstrated that exists a deterioration in the users experience when they are using P2P networks with some techniques of traffic localization.

We have prepared a network with a variable number of clusters, a representative number of total peers (600 and 1200), and a variable number of file size (10Mb and 20Mb). Also, we have connected the clusters in two different topologies, Ring Topology and Mesh Topology. This two topologies are opposite. Ring Topology has the less number of interconnection between clusters. On the other hand, Mesh Topology has the maximum number of interconnections between peers.

The conclusions that we have obtained for Ring Topology:

- Cluster_0 (contains the seeder) has the maximum priority in the network. The average time of this cluster is always the minimum.
- The delay that users experiment is directly proportional to the distance from their cluster to the cluster that contains the seeder.
- Increasing the number of cluster affects all the nodes. The farer you are from 0-node, the more time you expend to download it.
- Changing the file size does not affect the behavior of the results.
- Changing the number of total users does not affect the behavior of the results.

Traffic localization has as a priority to reduce the number of interconnection in the hole network. Ring Topology is a very good topology for do that. We can connect N-clusters with only N connections. On the other hand, when we apply this topology, we can see how appear a new parameter in our topology: number of hoops to the cluster 0. Also, we have demonstrated how important is this parameter in users' experience; farer you are from 0-node more priority you have and more delay users experiment (in other words, bad users' experience).

The conclusions that we have obtained for Mesh Topology:

- Average time of 0-hoop distance is constant for almost every condition (number of clusters, nodes, interconnections, etc.).
- The average time of 1-hoop distance clusters (every cluster except cluster that contains the seeder) decreases if the number of connections between clusters increases.

- There is no difference in average time of different clusters at 1-hoop distance. In ring topology, farer you are from node 0, more delay you have.
- Increasing the number of cluster does not affect average time too much. This is because the total number of interconnections between clusters is bigger. The file does not need to surround all the topology.
- The number of clusters that we can form with this topology is smaller than in other topologies. In our experiments, we have to change the parameter to measure the number of interconnections (rho).
- Changing the file size does not affect the behavior of the results.
- Changing the number of total users does not affect the behavior of the results.

As we already commented in Ring Topology, traffic localization is about decreasing the number of interconnections in the hole network. With Mesh Topology we have the maximum number of connections between clusters. So, *full* Mesh Topology is not the best option in this area. Also, if we increase the number of clusters, the number of interconnections increases in a factor bigger than in Ring Topology. On the other hand, with Mesh Topology we do not have any additional parameter like in Ring Topology -distance from one cluster to the cluster_0-. Also, in Mesh Topology, all the nodes of all the clusters (except cluster 0) has the same user experience.

In our work we are simulating just a few range of the multiple possibilities. For example, the number of users of every cluster are constant. In real world, cluster should be dynamic and different sized. Also, the bandwidth and delay of those peers should be different and more representative. If we want to simulate more precisely this kind of network, we should modify our *channels file* and include an average of connection for every country/company and make our simulator select different connections for different clusters simulating cluster by differences of clusters or companies.

Also, we have seen the relevance of the topology selected in network behavior. To obtain more accurate results we should try to construct new topologies in order to find the most efficient one or the less harmful structure for users delay.

There are many studies [5, 8, 22] that demonstrate the beneficial effects of traffic localization for ISP companies. It reduces the traffic intra and inter AS. It is more efficient connecting peers that are near. Those studies only focus on ISP services and needs. With our work we have demonstrate that not everything is positive in Traffic Localization. There are many factors that we have to take in mind before implement this technique.

Before the implementation of any kind of technique for traffic localization we have to study all the pros and cons. Now a days, there are not enough information about the negative effects of traffic localization. The development of P2P software and protocol should have to take in mind all the effects -positive and negative-, especially those effect that affect final user. We cannot implement a traffic localization technique that make users leave P2P networks.

P2P development need to take care of users experience. It would be very harmful for P2P networks an earlier and unstudied implementation of Traffic Localization.

References

[1] 3CX. http://www.voipforo.com/codec/codecs.php.

[2] BandaAncha. Ono filtering p2p traffic? www.bandaancha.eu.

[3] BitTorrent. http://www.bittorrent.com.

[4] Michael Howarth Chaojiong Wang, Ning Wang. A new approach for achieving traffic-exchange localization in p2p-based content distribution. University of Surrey, 2009.

[5] Cheng-Ying Li Chi-Jen Wu and Jan-Ming Ho. Improving the download time of bittorrent-like systems. Institue of Internet Science, 2007.

[6] Xizhi Zhu Dmitri Moltchanov. Quality issues and challenges in p2p streaming. Department of Communication Engineering,, 2009.

[7] Ph.D. Dr. Dmitri Moltchanov. Service quality in p2p streaming systems. Elsevier Editorial System(tm) for Computer Science Reviews, 2010.

[8] Vinay AggarwalObi AkonjangAnja Feldmann. Improving user and isp experience through isp-aided p2p locality. IEEE, 2008.

[9] 2 Guobin Shen1, Ye Wang1. Hptp: Relieving the tension between isps and p2p. MicrosoftResearchAsia,Beijing,P.R.China, 2007.

[10] http://mm.aueb.gr/. Bittorrent model for oversim. http://mm.aueb.gr/.

[11] http://wappy.ws/tag/BitTorrent. Bittorrent the highest percentage of internet use (spanish).

[12] IETF. http://datatracker.ietf.org/doc/draft-ietf-speermint-voip-consolidated-usecases/.

[13] Stephan Krause Ingmar Baumgart, Bernhard Heep. Oversim: A flexible overlay network simulation framework. Institute of Telematics, 2007.

[14] ITU-T. H.323 : Packet-based multimedia communications systems.

[15] Martin Stiemerling Jan Seedorf, Sebastian Kiesel. Traffic localization for p2p-applications: The alto approach. NEC Laboratories Europe, 2009.

[16] Vasileios P. Kemerlis Konstantinos Katsaros. A bittorrent module for the omnet++ simulator. Athens University of Economics and Business, 2008.

[17] Chao Liang; Yang Guo; Yong Liu. Is random scheduling sufficient in p2p video streaming? IEEE, 2008.

[18] Mircrosoft. http://research.microsoft.com/en-us/downloads/20d68689-9a8d-44c0-80cd-66dfa4b0504b/.

[19] Alberto Montresor. Peersim: A scalable p2p simulator. University of Trento, Italy, 2009.

[20] Mikko Vapa Niko Kotilainen. P2prealm – peer-to-peer network simulator. Department of Mathematical Information Technology, 2007.

[21] OverSim. http://groups.google.com/group/omnetpp-bittorrent.

[22] Cristian Grecu Partha Pratim Pande. Effect of traffic localization on energy dissipation in noc-based interconnect. SoC Research Lab, Department of Electrical and Computer Engineering, 2005.

[23] D. Moltchanov R. Dunaytsev, J. Peltotalo. P2P Live Streaming: Selected Systems. Tampere University of Technology, TLT-2626, PEER-TO-PEER NETWORKS, - edition, - -.

[24] Christian Scheideler and Anja Feldmann Vinay Aggarwal. Can isps and p2p users cooperate for improved performance? -, 2007.

[25] Riidiger Schollmeier. A definition of peer-to-peer networking for the classification of peer-to- peer architectures and applications. Institute of Communication Networks, Technische Universitat, 2002.

[26] www.ipoque.com. Internet study 2008/2009.

[27] Weishuai Yang and Nael Abu-Ghazaleh. Gps: A general peer-to-peer simulator and its use for modeling bittorrent. Department of Computer Science, Binghamton University, 2005.